Razvoj skladišta podataka i integracija podataka iz heterogenih izvora

Stanović, Vlaho

SVEUČILIŠTE U DUBROVNIKU
ODJEL ZA ELEKTROTEHNIKU I RAČUNARSTVO

VLAHO STANOVIĆ

RAZVOJ SKLADIŠTA PODATAKA I INTEGRACIJA PODATAKA IZ HETEROGENIH IZVORA

DIPLOMSKI RAD

Dubrovnik, svibanj 2021.

SVEUČILIŠTE U DUBROVNIKU
ODJEL ZA ELEKTROTEHNIKU I RAČUNARSTVO

RAZVOJ SKLADIŠTA PODATAKA I INTEGRACIJA PODATAKA IZ HETEROGENIH IZVORA

DIPLOMSKI RAD

Studij: Primijenjeno/poslovno računarstvo
Studijski smjer: Primijenjeno računarstvo
Kolegij: Napredni modeli i baze podataka
Mentor: izv. prof. dr. sc. Mario Miličević
Komentorica: Ines Obradović, mag. ing. comp.
Student: Vlaho Stanović

Dubrovnik, svibanj 2021.

SAŽETAK

U današnje doba, potreba za skladištenjem i obradom podataka je sve veća. Cilj skladištenja podataka je imati povijesne podatke nekog poduzeća na jednom mjestu te vršiti analize i poslovne odluke na temelju tih podataka. Ono što se često spominje kao izazov prilikom rada sa podacima je činjenica da su podaci međusobno udaljeni te se nalaze u različitim dokumentima ili bazama podataka. Nužnost za sjedinjenjem tih udaljenih podataka u jednu bazu postala je prioritet brojnim poduzećima današnjice. Jedna od osnovnih prednosti skladištenja podataka za poduzeća, odnosno klijente je mogućnost analize podataka. Analizom podataka omogućuje se kvalitetnije donošenje poslovnih odluka što doprinosi održavanju i razvoju poduzeća u današnjem konkurentnom poslovnom okruženju.

Tema ovog rada je proces skladištenja podataka i prikaz ovog procesa na primjeru izgradnje jednostavnog skladišta podataka. U prvom dijelu rada obrađuje se postojeća literatura na temu procesa skladištenja podataka te se navode i detaljno obrađuju svi koraci nužni za kreiranje i rad sa skladištem podataka. Nadalje, uspoređuju se dva pristupa skladištenju podataka; Inmonov i Kimballov model, te se detaljno govori o njihovim sličnostima i razlikama. U drugom dijelu rada obrađuje se studijski primjer koji za cilj ima približiti i objasniti izradu skladišta podataka na primjeru podataka iz fiktivnog poduzeća naziva Adventure Works Cycles. U ovom dijelu je prikazan cjelokupan proces skladištenja podataka, što znači da se počinje prikazom podatka u relacijskoj bazi podataka, a završava sa prikazom podataka pogodnih za analizu od strane krajnjih korisnika. Tehnologija i alati korišteni u ovom radu su Microsoftovi suvremeni alati koji omogućuju brzu i jednostavnu izgradnju skladišta podataka.

Ključne riječi: činjenična tablica, dimenzijska tablica, ETL proces, OLAP kocka, OLAP sustav, relacijska baza podataka, skladište podataka, zvjezdasti model

ABSTRACT

In today's world, there is a growing need for data warehousing and data processing. The goal of data warehousing is to have a company's historical data in one place and make analysis and business decisions based on these data. The challenge that is commonly mentioned when working with data is the fact that the data is remote from each other and resides in different documents or different databases. The need to merge the remote data into a single database has become a priority for many businesses today. One of the main benefits of data warehousing for businesses or clients is the ability to perform data analytics. Data analysis leads to making quality business decisions that contribute to the presence and growth of the company in today's competitive business environment.

The topic of this paper is the process of data warehousing and the illustration of this process using the example of building a simple data warehouse. In the first part of this paper, the existing literature on data warehousing is reviewed and all the steps required to create and work with data warehouse are mentioned and discussed in detail. Also, two approaches to data warehousing, Inmon's and Kimball's model, are compared and their similarities and differences are evaluated. The second part of the paper deals with a practical example, which aims to bring closer and explain the construction of a data warehouse using the data of a fictitious company called Adventure Works Cycles as an example. The whole process of data warehousing is shown, i.e. this part starts with the representation of the data in the relational database and ends with the representation of the data suitable for analysis by the end- users. The technology and tools that have been used in this paper are modern tools from Microsoft that allow for quick and easy construction of data warehousing.

Keywords: data warehouse, dimensional table, ETL process, fact table, OLAP cube, OLAP system, relational database, star model

SADRŽAJ:

SAŽETAK i
ABSTRACT ii
1. UVOD 1
2. SKLADIŠTE PODATAKA 3
2.1. Osnovni pojmovi 3
2.2. Kratka povijest pohrane i skladištenja podataka 4
2.3. Relacijske baze podataka 7
2.3.1. Relacijske baze podataka – osnovne značajke 7
2.4. Pristupi skladištenju podataka 10
2.4.1. Inmonov model 10
2.4.2. Kimballov model 11
2.5. Višedimenzionalni model skladišta podataka 12
2.5.1. Činjenične tablice 13
2.5.2. Dimenzijske tablice 13
2.6. Modeli skladišta podataka 14
2.6.1. Zvjezdasti model 14
2.6.2 Pahuljasti model 15
2.7. Usporedba transakcijske baze podataka i skladišta podataka 16
3. OLAP SUSTAV 18
3.1. Višedimenzionalni modeli 18
3.1.1. Vrste operacija nad višedimenzionalnim modelom 18
3.2. Tablični model 23
3.3. Usporedba modela 24
3.4. ETL proces 25
3.5. Uloge i odgovornosti ETL stručnjaka 27
3.6. Problemi kod ETL procesa 27
4. IMPLEMENTACIJA SKLADIŠTA PODATAKA 28
4.1. Korištene tehnologije 28
4.2. Definicije osnovnih pojmova 30
4.3. Namjena i proces implementacije skladišta podataka 28
4.4. Odabir podataka iz relacijske baze 31
4.5. Izgradnja skladišta podataka 33
4.5.1. Izrada dimenzijskih tablica 33
4.5.2. Punjenje dimenzijskih tablica podacima 34
4.5.3. Kreiranje i punjenje vremenske dimenzije 37
4.5.4. Izrada činjenične tablice 41

4.5.5. Punjenje činjenične tablice 42
4.5.6. Prikaz upita u skladištu podataka 55
5. PRIKAZ REZULTATA U VIŠEDIMENZIONALNOM MODELU 58
6. ZAKLJUČAK 61
LITERATURA 62

1. UVOD

Doba digitalizacije i sve veće prisutnosti brojnih *online* rješenja, doveli su do pojave da organizacije, poslovni odjeli i sami krajnji korisnici (engl. *end-users*) mogu bez značajnih poteškoća razvijati i dalje raditi na svojim programima odnosno aplikacijama. Takva sveprisutna praksa dovela je do pojave velikog broja manjih baza podataka u firmama. Ono što se pokazalo kao izazov u ovakvoj praksi je činjenica da su ove baze heterogene i korisnik koji radi na njima često ima poteškoće koje uzrokuju razdijeljeni i udaljeni podaci. Ovaj izazov u poslovnom svijetu tražio je od IT stručnjaka jednostavno i logičko rješenje tj. pojavila se potreba za organiziranjem i spajanjem ovih razdvojenih podataka u jednu homogenu, smislenu cjelinu. Krajnji rezultat integracije ovakvih podataka je zapravo skladište podataka. U skladištu podataka nalaze se povijesni podaci poslovanja nekog poduzeća, a pristup tim podacima je jednostavan i brz. Osim pristupa samim podacima, skladište podataka omogućuje krajnjim korisnicima da uz pomoć raznih sofisticiranih tehnoloških alata, intuitivno i jednostavno vrše razne analize podataka koje mogu doprinijeti napretku samog poslovanja pojedinog poduzeća.

Cilj ovog rada je staviti fokus na skladište podataka te jednim suvremenim primjerom prikazati izgradnju skladišta podataka koje vuče podatke iz različitih heterogenih izvora. Važno je naglasiti da će se u ovom radu prikazati cjelokupan i detaljan proces skladištenja podataka, što uključuje kao prvi korak prikaz podatka u relacijskoj bazi podataka do konačnog koraka procesa, a to je prikaz podataka u završnom izvještaju napravljenom u programu Excel. Na ovaj način, prikazom i radom na praktičnom primjeru, steći će se osnovno znanje i bolje shvatiti sam proces skladištenja podataka.

Kako bi se svrha i ciljevi ovog rada što bolje ostvarili, u radu će se koristiti Microsoftove tehnologije i alati kao što su SSMS (*SQL Server Management Studio*), SSIS (*SQL Server Integration Services*) i SSAS (*SQL Server Analysis Services*). Navedene tehnologije su posvećene radu s podacima, veoma su intuitivne te jednostavne za korištenje.

Ovaj rad se sastoji od četiri osnovna poglavlja. Prvo poglavlje iznosi teorijski pregled skladišta podataka i njegovu usporedbu sa relacijskom bazom podataka. Nadalje, prvo poglavlje opisuje dva najvažnija modela pristupa skladištenju podataka, Inmonov i Kimballov model, te opisuje dva najvažnija modela skladišta podataka, zvjezdasti i pahuljasti model.

U drugom poglavlju ovog rada govori se o OLAP sustavu te se iznose najvažnije karakteristike ovog sustava. U ovom dijelu također se dotiče tema ETL procesa i objašnjava zašto je ovaj proces najvažniji dio skladištenja podataka. Treće poglavlje predstavlja studijski primjer koji za cilj ima kreiranje skladišta podataka na primjeru podataka iz fiktivnog poduzeća. Nadalje, u ovom dijelu prikazati će se ETL proces tj. proces punjenja skladišta podataka potrebnim podacima. Četvrto poglavlje prikazuje izradu kocke i prikazuje završni korak procesa

skladištenja podataka koji za cilj ima kreiranje završnog izvještaja kojeg koriste krajnji korisnici u svrhu poslovnih analiza i odlučivanja.

2. SKLADIŠTE PODATAKA

Prema definiciji, skladište podataka (engl. *data warehouse*) služi kao potpora u radu sa velikim količinama podataka. Ideja je da se podaci iz manjih tzv. operativnih baza izdvoje i pohrane u posebno osmišljene baze (skladište podataka) kako bi se ti svi podaci mogli koristiti za detaljnije i složenije analize [1].

Skladištenje je u suštini integracija podataka neke organizacije u jedinstvenu bazu podataka. Cilj skladištenja podataka je omogućiti jednostavno analiziranje ili pregled integriranih podataka. Model pohranjivanja podataka je pogodan za analizu te se na taj način podaci mogu puno brže pregledavati, a često se podaci jedino kroz skladište podataka mogu i vidjeti, najčešće zbog složenosti ili pristupa samim podacima [2].

2.1. Osnovni pojmovi

Ideja skladišta podataka, tj. spremanja podataka na pogodan način za analizu i izvještavanje, rodila se još šezdesetih godina, ali tek krajem prošlog stoljeća doživljava procvat. Preduvjeti za ovakav nagli razvoj bili su u tome da su se poduzeća informatizirala, ali nisu bila u mogućnosti pregledavati podatke ili su ih pregledavali prekasno gledajući s poslovne pozicije. Poduzeća su prepoznala važnost pravovremene informacije za prednost na tržištu, a razvoj industrije, posebno računala te programske potpore uz pojeftinjenje osnovnih proizvoda za pohranjivanje i obrađivanje podataka, uvelike je doprinio zamahu i procvatu skladištenja podataka [2].

Otac skladišta podataka, Bill Inmon, opisao je skladište podataka kao tematski orijentiran, integriran, vremenski ovisan i postojan skup podataka koji služi kao potpora u procesu odlučivanja [2]. Slijedi kratki opis svake od ovih karakteristika. Prva od navedenih karakteristika skladišta podataka je tematska orijentiranost. Ova karakteristika se odnosi na činjenicu da je skladište podataka orijentirano na podatke zanimljive za analizu, tj. aktivnosti od velikog značaja za poduzeće. Sljedeća važna karakteristika skladišta podataka je integriranost. Integriranost se odnosi na podatke tj. na činjenicu da podaci u skladištu podataka moraju imati jedinstven format radi konzistentnosti. To nije lako osigurati s obzirom na to da podaci najčešće dolaze iz brojnih različitih izvora. Sljedeća karakteristika se odnosi na vremensku komponentu. Podaci su vremenski ovisni, što znači da se u transakcijskim bazama konstantno mijenjaju i brišu. Skladište podataka predstavlja skup povijesnih podataka, od kojih je svaki podatak bio suvremen u jednom trenutku. Posljednja karakteristika skladišta podataka je postojanost podataka. Podaci u skladištu podataka su nepromjenjivi, osim ako su uneseni pogrešno. Nad skladištem podataka u pravilu bi se trebale izvoditi samo dvije operacije -

periodičko učitavanje podataka, tj. punjenje podacima, te čitanje podataka. S obzirom na to da se podaci nikada ne mijenjaju niti brišu, nema opasnosti od nekonzistentnosti podataka i otvara se mogućnost korištenja višedimenzionalnih modela podataka.

2.2. Kratka povijest pohrane i skladištenja podataka

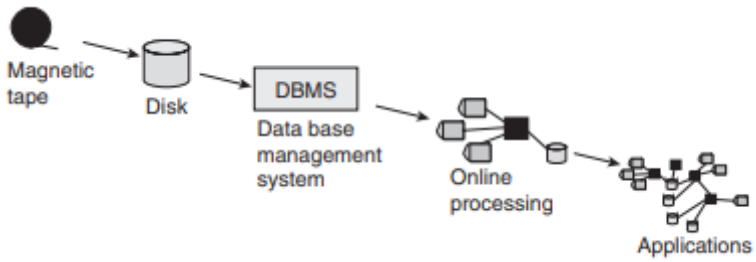
Skladište podataka je dizajnirano kako bi podržalo proces donošenja odluka kroz prikupljanje, spajanje, analiziranje i istraživanje podataka. Skladište podataka može se koristiti u pojedinačnim poslovnim područjima nekog poduzeća, ovisno o potrebi, kao što su prodaja, financije, ljudski resursi i ostala područja. Važno je spomenuti i da danas skladišta podataka čine važnu komponentu poslovne inteligencije (engl. *business intelligence*) [3].

Inmon, Strauss i Neushloss u svojoj knjizi naziva „*DW 2.0: The Architecture for the Next Generation of Data Warehousing"* govore o povijesti skladišta podataka i opisuju razloge zbog kojih je došlo do razvoja skladišta podataka [4]. Na samom početku, podaci su se čuvali na bušenim karticama odnosno papirnatim vrpcama. Takva vrsta čuvanja, odnosno pohrane podataka je bila iznimno skupa i limitirana. Novi trenutak u povijesti skladištenja podataka je došao sa razvojem i uvođenjem magnetne vrpce/trake. Specifično za magnetnu traku je bilo to da je mogla pohraniti velike količine podataka i to vrlo jeftino. Važna značajka magnetne vrpce je bila i ta da su se podaci na njoj mogli mijenjati i samim time, ona je označila veliki iskorak za skladištenje podataka. Nedostatak magnetne vrpce je bio taj da se podacima moglo pristupati samo sekvencionalno (jedno iza drugog) tj. kako bi osoba došla do 1% podataka, 100% podataka se moralo pročitati i moralo se njima fizički pristupiti.

Sljedeći veliki iskorak u povijesti skladištenja podataka dogodio se izumom diska i pohranom podataka na disku. Kod ovakve pohrane podataka, podacima se moglo direktno pristupiti, podaci su se mogli zapisivati i mijenjati, i ono što je najvažnije, podacima se moglo masovno pristupati za razliku od podataka zapisanih na magnetnoj vrpci. U nastavku na izum diska i skladištenje podataka na disku, dogodio se još jedan napredak, a to je izum softvera pod nazivom DBMS (engl. *data base management system*). Ovaj softver je omogućio upravljanje bazom podataka odnosno upravljanje pohranjenim podacima. Najveća prednost skladištenja podataka na disku je bila činjenica da su se podaci mogli pronaći i pregledati vrlo brzo te samo postojanje DBMS-a na disku koji je još više olakšao pristup podacima.

Napredak u skladištenju podataka pratio je napredak u tehnologiji, pa se sljedeći korak u povijesti skladišta podataka veže za *online* aplikacije. *Online* aplikacije su ovisile o računalu kako bi se podacima moglo brzo pristupiti. Primjeri ovih aplikacija su: bankomati, aplikacije za rezervacije avionskih karata, aplikacije koje služe za nadzor proizvodnje, aplikacije koje

koriste službenici u bankama itd. *Online* aplikacije su ubrzo postale veoma popularne i prerasle su u „ispreplетene“ aplikacije. Slika 1 ilustrira opisanu povijest skladištenja podataka.

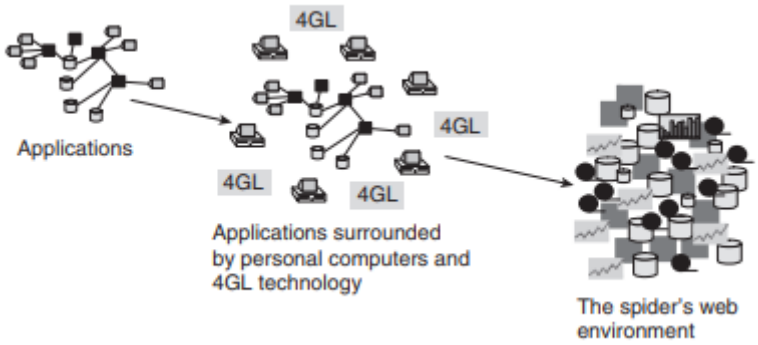


Slika 1. Razvoj skladišta podataka [4]

Novi iskorak u povijesti skladištenja podataka dogodio se pojavom osobnih računala i 4GL (engl. *fourth- generation language*) tehnologije. Glavna ideja iza 4GL tehnologije je bila u tome da se razvoj sustava i programiranje toliko pojednostavi i olakša da svi u tome mogu sudjelovati (bilo tko može jednostavno i bez poteškoća sudjelovati). U ovom razdoblju, ideja je bila da se krajnji korisnik „oslobodi“ i da može sam, bez prevelike podrške od strane IT odjela, pristupiti i raditi sa podacima. Na ovaj način, krajnji korisnik bi bio jako zadovoljan te nije trebalo dugo vremena da osobna računala i 4GL tehnologija uđu u brojne svjetske korporacije. Ovaj pristup je donio brojne prednosti, ali i izazove krajnjim korisnicima. Primijetili su da nije dovoljno samo imati pristup podacima kako bi mogli donijeti dobru odluku. Izazovi koji su mučili krajnje korisnike bili su sljedeći:

a) ako podaci nisu točni, mogu biti veoma zavaravajući i u krajnjem slučaju opasni;
b) nepotpuni (djelomični) podaci su beskorisni;
c) podaci koji nisu pravovremeni nisu ni poželjni;
d) ako postoji više verzija istih podataka, lako se može donijeti pogrešna odluka ukoliko krajnji korisnik gleda pogrešnu ili nepotpunu verziju podataka;
e) podaci bez popratne dokumentacije nisu vrijedni.

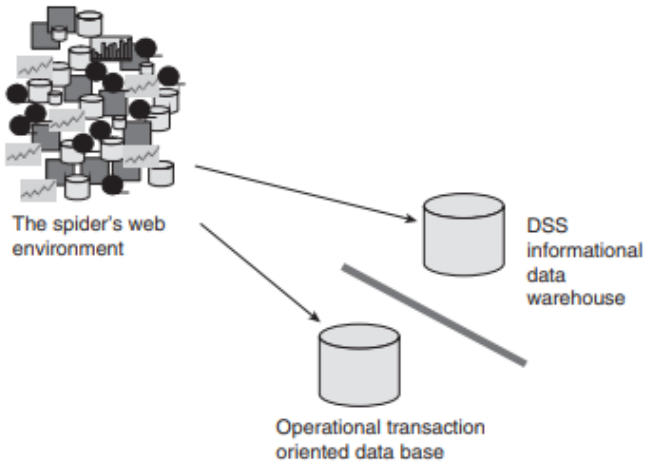
Rezultat ovakve prakse je bio potpuni kaos. Taj kaos se često zove paukova mreža (engl. *spider's web environment*). Tako se naziva jer postoje mnoge putanje podataka koje idu u različitim smjerovima i zbog toga podsjećaju na paukovu mrežu. Slika 2 ilustrira razvoj paukove mreže u korporacijama.



Slika 2. Razvoj paukove mreže [4]

Paukova mreža je bila jako loša za korporacije i nije postojao način da se paukova mreža učini korisnom i funkcionalnom. Frustracija koju je paukova mreža stvorila kod krajnjih korisnika, IT stručnjaka i menadžera u korporacijama, dovela je do razvoja drugačije arhitekture u čijem je središtu bilo postavljeno skladište podataka.

Skladište podataka predstavlja veliki napredak u razmišljanju IT stručnjaka. Prije samog razvoja skladišta podataka, smatralo se da baza treba služiti svim vrstama podataka i da može poslužiti za bilo koju svrhu. Međutim, razvojem skladišta podataka postalo je jasno da postoje mnoge vrste baza podataka. U ovom prijelazu, kao što slika 3 ilustrira, shvatilo se da se podaci dijele u različite vrste podatkovnih baza i da je nužno razviti novi sustav u kojem će skladište podataka biti odvojeno od transakcijske baze.



Slika 3. Podjela podataka u različite vrste baza podataka [4]

Foote [3] govori o tome kako se sa razvojem skladišta podataka počela događati i akumulacija velikih podataka (engl. *big data*) te uvelike utjecati na potrebu za razvojem računala, interneta, mobitela (engl. *smartphone*) i Interneta stvari (engl. *internet of things)* koji bi služili kao izvori podataka. Društveni mediji i kreditne kartice su također bili od značajne koristi u ovom periodu.

2.3. Relacijske baze podataka

Za bolje razumijevanje skladišta podataka i njegovih osnovnih značajki, u ovom dijelu je neophodno spomenuti i usporediti relacijsku bazu i skladište podataka.

2.3.1. Relacijske baze podataka – osnovne značajke

Relacijska baza podataka je skup međusobno povezanih neredundantnih podataka. Podaci unutar baze podataka opisuju sva pojavljivanja entiteta, veza te njihovih atributa. Veza između podataka je opisana u shemi baze podataka tj. vizualni prikaz modela baze podataka. Transakcijski sustavi ili relacijske baze podataka sadrže trenutne i detaljne podatke. Podaci se često mijenjaju i sustav je orijentiran prema dnevnim operacijama i vođenju poslovnog sustava. Izuzetno je važna konstantna raspoloživost, obzirom da na sustavu radi veliki broj operativnih korisnika na dnevnoj bazi. Težište je na pohranjivanju podataka. Tehnologija baza podataka nastoji provesti sljedeće ciljeve [5]:

a) Fizička neovisnost podataka

Fizička neovisnost podataka podrazumijeva razdvojenost logičkog dijela baze i njene fizičke građe. To znači da premještanje podataka na drugo fizičko mjesto neće utjecati na aplikacije koje koriste podatke.

b) Logička neovisnost podataka

Logička neovisnost podataka podrazumijeva razdvajanje globalne i lokalne logičke definicije za aplikaciju. Dakle, ako se promijeni logička definicija umetanjem novog zapisa ili veze, ta promjena ne bi smjela zahtijevati promjene u postojećim aplikacijama.

c) Fleksibilnost pristupa podacima

Fleksibilnost pristupa podacima podrazumijeva mogućnost da se korisnik može slobodno kretati po podacima te uspostavljati veze među podacima kako on misli da je najbolje. Prije razvoja relacijskih baza, korisnik je mogao pristupati podacima jedino onim redoslijedom koji je bio prethodno isplaniran prilikom projektiranja baze.

d) Istovremeni pristup podacima

Istovremeni pristup podacima podrazumijeva mogućnost korištenja baze od strane više korisnika koji moraju imati osjećaj da su jedini korisnik baze u tom trenutku.

e) Integritet podataka

Integritet podataka znači očuvanje podataka, posebice u trenucima u kojima dolazi do greške u aplikaciji ili istovremene aktivnosti korisnika.

f) Mogućnosti oporavka

Mogućnost oporavka označava postojanje pouzdane zaštite baze u slučaju kvara, bilo da je riječ o kvaru hardvera ili nekoj greški u radu softvera.

g) Zaštita korištenja

Zaštita korištenja za cilj ima ograničavanje prava korisnicima prilikom korištenja baze, ovisno o tome koje funkcionalnosti korisnik smije koristiti ili koje podatke korisnik smije vidjeti.

h) Zadovoljavajuća brzina pristupa

Zadovoljavajuća brzina podrazumijeva da se operacije nad podacima obavljaju toliko brzo koliko aplikacija koja koristi te podatke zahtijeva. Na brzinu se može utjecati ili boljim, odnosno jačim hardverom, ili pametnijim pisanjem koda.

i) Mogućnost podešavanja i kontrole

Mogućnost podešavanja i kontrole je posao administratora baze podataka. Njegov posao je zapravo briga o pojedinoj bazi podataka. Neki od poslova na bazi nakon njezine implementacije su pohranjivanje podataka, reguliranje prava korisnika, praćenje performansi te mijenjanje određenih parametara. Također, projekti i aplikacije se s vremenom mijenjaju kao i baze nad kojima aplikacije rade te je stoga potrebno mijenjati logičku strukturu baze.

Važno je napomenuti i objasniti arhitekturu baze podataka kako bi se bolje shvatilo kako baza podataka funkcionira. Sama arhitektura baze podataka sastoji se od tri razine apstrakcije tj. tri sloja [5]:

a) Fizička razina (engl. *physical level*)

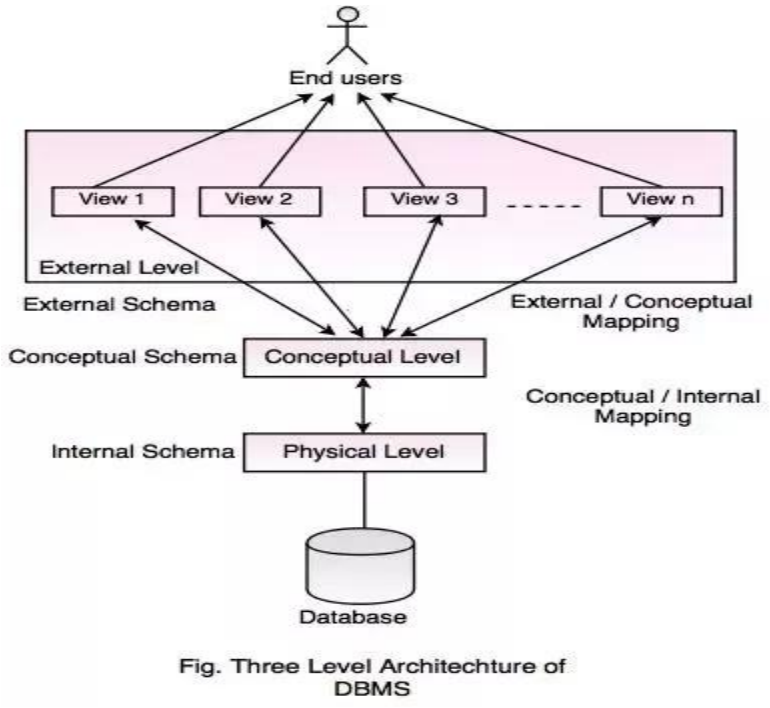
Fizička razina opisuje raspored podataka u memoriji. Sistemski programeri jedini su koji to mogu vidjeti te je ovaj aspekt prilično apstraktan. U većini slučajeva IT stručnjaci ne znaju i nemaju potrebe znati što se događa na ovoj razini.

b) Globalna logička razina (engl. *global conceptual level*)

Globalna logička razina je logička struktura cijele baze. Logička definicija baze podataka naziva se shema baze podataka. Shema je najčešće dijagram i u skladu je s modelom. Pristup globalnoj logičkoj razini imaju projektant ili administrator baze.

c) Lokalna logička razina (engl. *local conceptual level*)

Lokalna logička razina odnosi se na dio baze kojeg koristi pojedina aplikacija. Taj aspekt vidi korisnik ili aplikacijski programer. Svaki pojedini zapis lokalne logičke razine naziva se podshema ili pogled (engl. *view*). Najvažnije, pogled definira preslikavanje iz globalnih podataka i veza u lokalnu logičku razinu. Slika 4 prikazuje raspodjelu arhitekture baze podataka prema gore opisanim razinama.



Slika 4. Arhitektura baze podataka [6]

U životnom ciklusu baze podataka, prvi korak je implementacija baze podataka u poduzeće. To je složen zadatak i implementacija se tretira kao projekt koji se može podijeliti u pet faza opisanih u nastavku [7].

Prva faza životnog ciklusa baze podataka je analiza potreba. Analiza potreba se odnosi na proučavanje tokova informacija te na identifikaciju podataka koje treba pohranjivati i veze između njih. Analizom je također potrebno analizirati tj. definirati transakcije odnosno operacije koje će se obavljati nad bazom podataka, obzirom da to može utjecati na završni oblik baze. Transakcije je potrebno planirati imajući na umu njihov opseg, kompleksnost i najvažnije, njihove performanse. Dokument u kojem je zapisan rezultat analize potreba naziva se „Specifikacija potreba“.

Druga faza odnosi se na modeliranje podataka. Prema definiciji, modeliranje podataka se odnosi na izradu globalne sheme na temelju specifikacije potreba. Nadalje, shema se normalizira tako da se zadovolje zahtjevi kvalitete te se dodatno modificira imajući na umu utjecaj na performanse. Posljednje se iz globalne sheme vade pod-sheme za pojedine grupe korisnika ili aplikacija.

Treća faza podrazumijeva implementaciju. Na osnovi prethodnih koraka te uz pomoć DBMS-a (engl. *data base management system)*, baza podataka se fizički kreira na računalu. Svaki DBMS sadrži posebne parametre koje je potrebno podesiti tako da se osigura kvalitetan rad bitnih transakcija. Ova faza također se sastoji od inicijalnog punjenja podacima.

Četvrta faza životnog ciklusa baze podataka uključuje testiranje. Testiranje se odnosi na pokusni rad s bazom i provjera zadovoljava li baza podataka sve zahtjeve i kriterije. Detaljno se radi s bazom podataka da bi se otkrile potencijalne greške koje su se potkrale u prethodnim koracima. U novije vrijeme, u prethodnom koraku implementacije, kreira se isključivo testna baza na kojoj se testira i traže greške. Kada su sve greške popravljene i baza podataka je spremna za redoviti rad, ponovno se vraća proces na implementaciju i testiranje na produkciji. Peta faza se odnosi na održavanje. Održavanja se događaju nakon ulaska baze u redovitu upotrebu. Održavanje je posao administratora baze podataka i u njegovu domenu spada popravljanje grešaka koje nisu otkrivene tokom testiranja, uvođenje promjena, te podešavanje parametara radi performansi. Održavanje je također konstantno praćenje rada same baze (engl. *monitoring*).

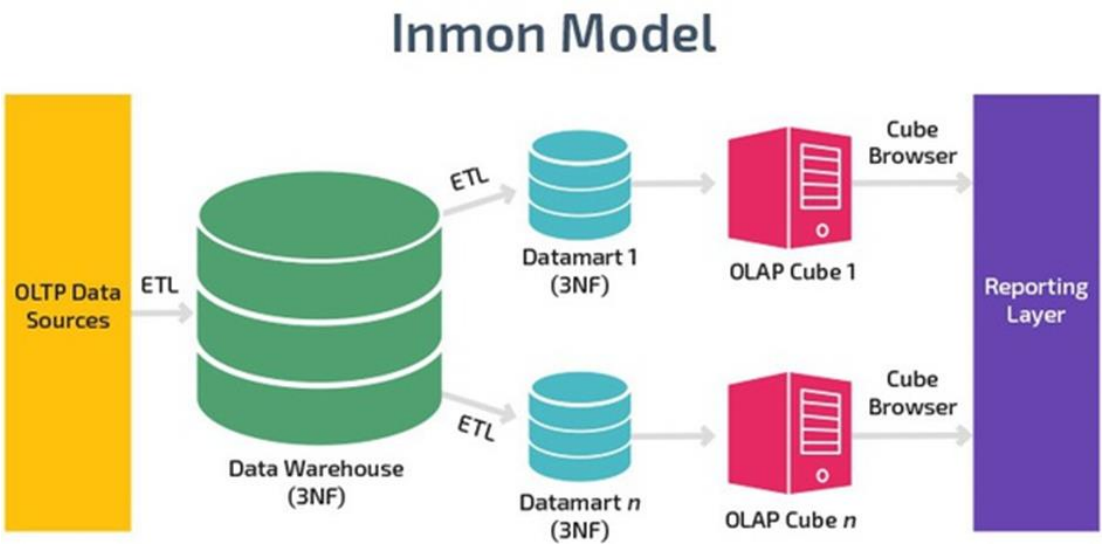
2.4. Pristupi skladištenju podataka

Prilikom razvoja skladišta podataka potrebno je razumjeti dva glavna modela za razvoj skladišta te shvatiti njihove sličnosti i razlike [8]. Za što bolje razumijevanje dvaju pristupa skladištenju podataka, potrebno je objasniti pojam tržnice podataka (engl. *data mart*). Tržnica podataka je struktura podataka specifična za skladišta podataka. Tržnica je zapravo manje skladište podataka unutar većeg skladišta podataka. Orijentirana je prema specifičnom poslovnom području npr. financijama ili poslovnom timu kao što su administratori baza podataka. To omogućuje da svaki odjel koristi, manipulira ili razvija vlastite podatke. Razlozi za kreiranje tržnice podataka su brz i olakšan pristup često potrebnim podacima, jednostavno kreiranje, manja cijena u usporedbi sa kreiranjem kompletnog skladišta podataka te činjenica da tržnice sadrže samo ključne i potrebne podatke [8].

Kada je riječ o skladištu podataka, razlikujemo Inmonov i Kimballov model skladišta podataka. Njihove karakteristike i specifičnosti su opisane u nastavku.

2.4.1. Inmonov model

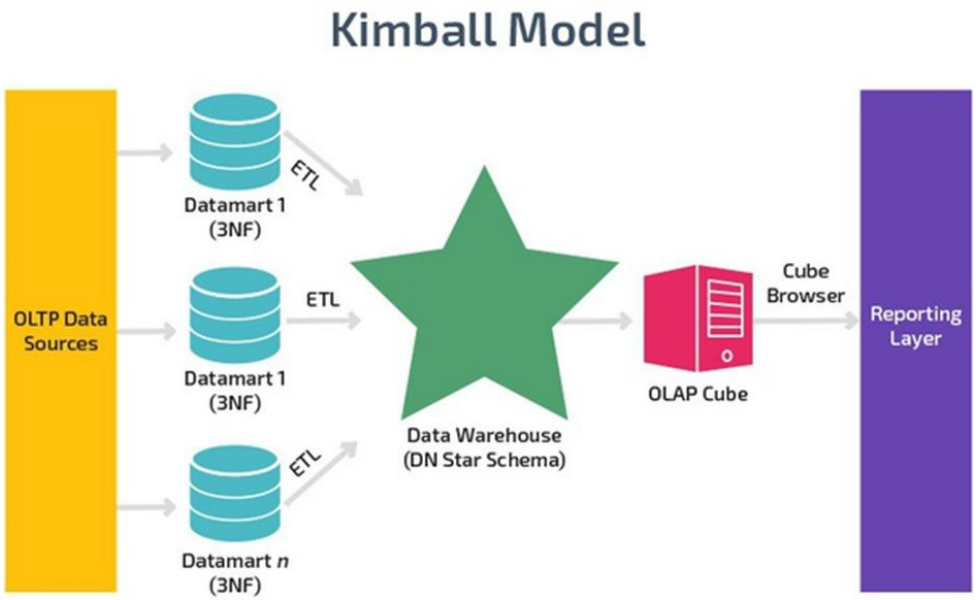
Inmonov model je model skladišta koji zastupa razvoj "od vrha prema dolje" (engl. *"top-down"*). Inmonov model skladišta podataka poštuje iste standarde kao i transakcijski sustavi. Razlog tome je što Inmon promatra sve podatke nekog poduzeća kao cjelinu te je potrebno imati sličnosti između različitih baza. Skladište podataka je normalizirano, a za pojedine upite, tj. pitanja koja će korisnici često postavljati, kreiraju se manji modeli [8]. Slika 5 prikazuje tok podataka, odnosno skladište podataka prema Inmonovom modelu.



Slika 5. Inmonov model skladišta podataka [9]

2.4.2. Kimballov model

Kimballov model je model skladišta podataka koji zastupa pristup "od dna prema gore" (engl. "*bottom-up*") te koristi dimenzionalno modeliranje svojstveno skladištima podataka. Kimballov model i prijedlog je da se za svaki poslovni proces kreira tržnica podataka te se na kraju svi ti različiti podaci povežu u skladište podataka [8]. Slika 6 ilustrira skladište podataka prema Kimballovom modelu.



Slika 6. Kimballov model skladišta podataka [9]

Inmonov i Kimballov model imaju dosta sličnosti, ali i razlika. Kada govorimo o sličnostima važno je sagledati ETL (engl. *extract, transform, load*) proces i vremensku dimenziju kao važne sličnosti. Vremenska dimenzija se smatra najvažnijom karakteristikom skladišta podataka. To je logično s obzirom na to da je prvotna ideja iza skladišta podataka potpora pri poslovnom odlučivanju. Pomoću vremenskog atributa, krajnji korisnici mogu uspoređivati npr. prodaju proizvoda na godišnjoj, mjesečnoj, dnevnoj razini te u ovisnosti je li bio vikend, praznik i slično [10].

S druge strane, ETL proces je ključan proces za skladište podataka zbog osiguravanja očuvanja integriteta podataka. U oba modela, podaci se izvlače iz transakcijskih sustava u skladište podataka (Inmon) ili u tržnicu podataka (Kimball) [10].

Međutim, prisutne su i brojne razlike između Inmonovog i Kimballovog modela. Ipak, najveći fokus je stavljen na razlike u modeliranju, metodologiji razvoja te arhitekturi samog skladišta. Kada je riječ o razlikama u metodologiji razvoja i arhitekturi, važno je naglasiti da je Inmonov model vrlo kompliciran i namijenjen IT stručnjacima. Samim time, izgradnja je duga i skupa. S druge strane, Kimballov model je jednostavniji te dopušta krajnjim korisnicima da se priključe razvoju. Izgradnja sustava je brza, ali dolazi do problema kod integracije zbog veće mogućnosti zalihosti i nekonzistentnosti [11].

Kada govorimo o razlikama u modeliranju podataka, Kimball temelji modeliranje podataka na procesima, tj. definira model na temelju odnosa između podataka u poslovnim procesima. Takav pristup je primjeren za višedimenzionalni model podataka jer se na temelju procesa odlučuje o činjeničnim i dimenzijskim tablicama. Krajnji korisnici mogu uz pomoć raznih alata za višedimenzionalno modeliranje sudjelovati u izgradnji skladišta podataka. Za razliku od Kimballovog, Inmonov model se temelji na podacima te se služi tradicionalnim načelima modeliranja podataka, za koje je potrebno predznanje i stoga je tu potrebno znanje i iskustvo IT stručnjaka [11].

2.5. Višedimenzionalni model skladišta podataka

Višedimenzionalno oblikovanje je logičko oblikovanje za predstavljanje podataka u jednostavnom, intuitivnom obliku pogodnom za učinkovit pregled podataka. Već u samim počecima razvoja relacijskih baza podataka, uočene su manjkavosti takvog modela pri pregledu podataka. Stotine relacija u većim transakcijskim bazama čini ih previše složenima za pregled. Razlog tome je činjenica što tada nije postojalo grafičko sučelje koje bi relacijski model učinilo intuitivnim i preglednim krajnjem korisniku te programska potpora ne može prolaziti kroz takve

relacijske modele bez ozbiljnog utjecaja na performanse [12]. Tablice unutar skladišta podataka dijele se na dvije glavne vrste, a to su dimenzijske i činjenične tablice.

2.5.1. Činjenične tablice

Činjenica (engl. *fact*) ili činjenične tablice su središnje točke skladišta podataka ili manjih tržnica podataka. Činjenice su zapravo one informacije koje će pomoći pri analizi i donošenju poslovnih odluka. Činjenične tablice su ključne tablice u višedimenzionalnom modelu, a u njih se pohranjuju numeričke mjere poslovanja [7]. U osnovi, činjenična tablica se sastoji od ključeva dimenzijskih tablica i njezinih mjera.

Najvažnije svojstvo relacijskih baza podataka - normaliziranost, također koristimo pri izgradnji činjeničnih tablica. Razlog tome je što činjenične tablice imaju veliki broj zapisa te pri tom sadrže samo numeričke atribute koji zauzimaju malo prostora na disku [7].

Mjere (engl. *measure*) u činjeničnoj tablici su jedan ili više numeričkih atributa koji uz kombinaciju stranih ključeva definiraju neki zapis. Pojednostavljeno, mjere su brojevi koji opisuju određeni poslovni proces (cijena, prodaja, plaća, itd.). S obzirom da krajnje korisnike prilikom analize i donošenja poslovnih odluka rijetko zanima samo jedan zapis (jedna ntorka), kod obavljanja upita često se obavljaju agregatne funkcije (sumiranje, prosjek, prebrojavanje itd.) pa se mjerom mogu nazvati i rezultati tih agregatnih funkcija [7].

2.5.2. Dimenzijske tablice

Dimenzijska tablica opisuje objekt koji sudjeluje u procesu koji se prati u skladištu podataka. Takve tablice daju kontekst te daju smisao, odnosno opisuju činjenice u činjeničnoj tablici. Dimenzijske tablice se sastoje od primarnog ključa, tj. ključa koji je spremljen u činjeničnim tablicama kao veza između tablica i atributa. Tipično, dimenzije nastaju spajanjem više relacijskih tablica, imaju manji broj zapisa, ali veliki broj atributa. Skupe su, odnosno zauzimaju puno prostora na disku [12].

Jedna od najvažnijih vrsta dimenzijske tablice je vremenska dimenzija. Većina poslovne analize se svodi na usporedbu poslovnih rezultata iz različitih perioda dana, mjeseca ili godine. Najbolji primjer su poduzeća koja se bave prodajom. Usporedba prodaje od pojedinog jutra do sljedećeg jutra, prodaja vikendom, prodaja oko blagdana, važne su informacije za poduzeće kako bi podiglo vlastitu konkurentnost. Iz tog razloga nije čudno što se vremenskoj dimenziji podataka

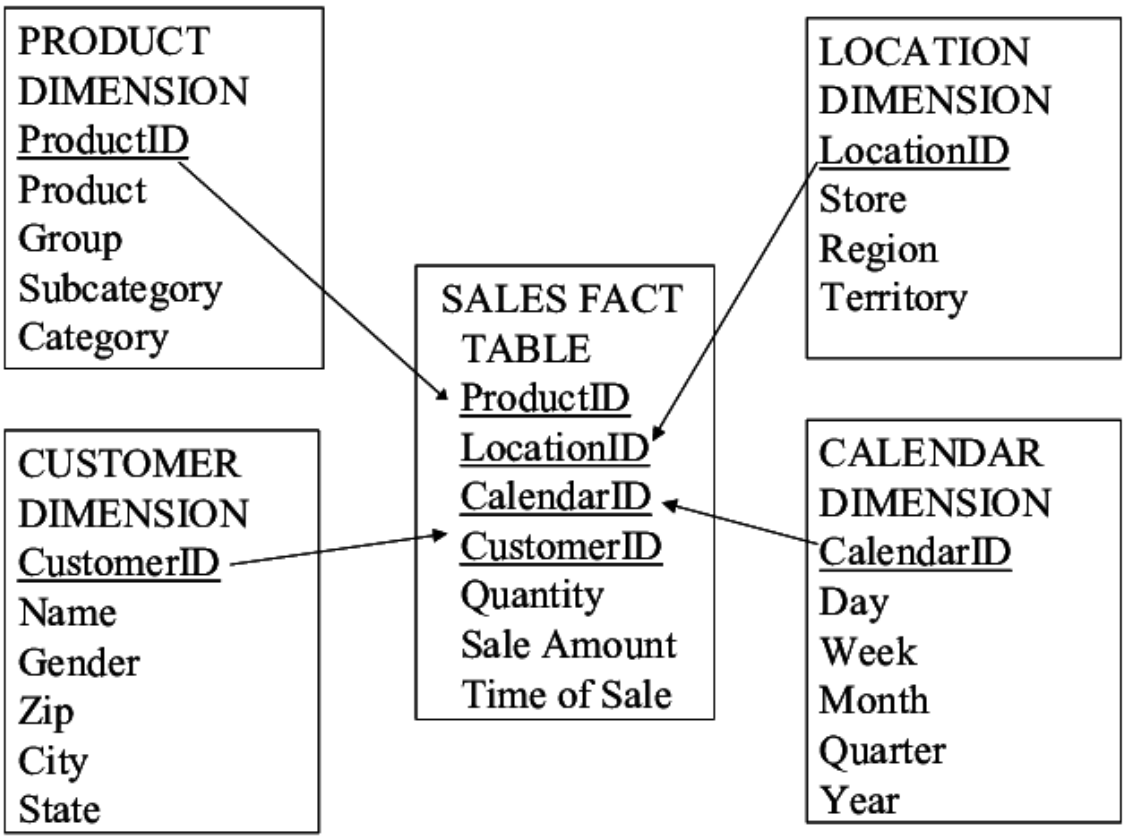
posvećuje posebna pažnja kreiranjem posebnih tablica za opis vremenskog konteksta. Razine zrnatosti takvih tablica treba prilagoditi poslovnim zahtjevima. Najčešća zrnatost je do jednog dana, tj. u tablici je veza jedan redak (jedna ntorka = jedan dan). Moguće je dovesti razinu zrnatosti do sekunde, ali kod takvog praćenja podataka pojedini upiti mogu vratiti i po više stotina milijuna podataka. Iz tog razloga izrazito je bitno prilikom izgradnje skladišta podataka razumjeti poslovne zahtjeve klijenta [12].

2.6. Modeli skladišta podataka

Modeli skladišta podataka se dijele u dvije najvažnije vrste, a to su zvjezdasti i pahuljasti model skladišta podataka. U nastavku slijede najvažnije značajke ova dva modela.

2.6.1. Zvjezdasti model

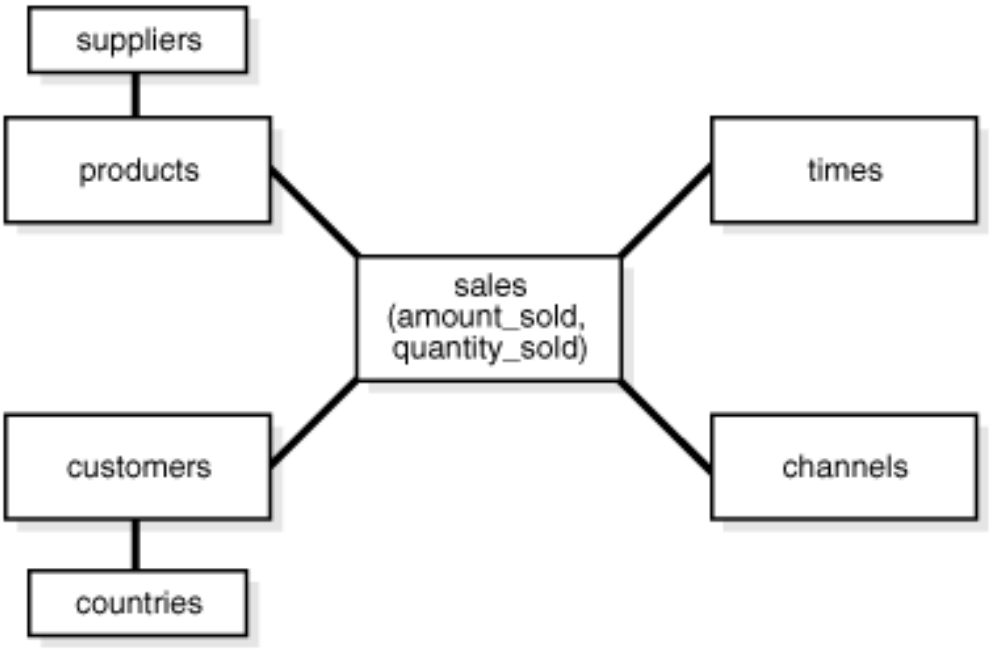
Zvjezdasti model je model koji sadrži jednu činjeničnu tablicu te više manjih dimenzijskih tablica. Vrlo je jednostavan, što dovodi do mnogih prednosti, poput smanjenja vremena potrebnog za izvođenje upita, jer ovakav model odgovara optimizatoru upita. Pregledan je te jednostavno pretražuje podatke. Ima predvidljivu strukturu te samim time razni alati za pisanje izvještaja mogu biti učinkovitiji u svojoj obradi znajući da su na poslužitelju podaci u zvjezdastom modelu [8]. Slika 7 prikazuje primjer skladišta podataka prema zvjezdastom modelu.



Slika 7. Zvjezdasti model skladišta podataka [13]

2.6.2. Pahuljasti model

Pahuljasti model je zapravo rezultat dekompozicije jedne ili više dimenzijskih tablica. U praksi se rijetko pojavljuje. Pahuljasti model daje dobru vizualizaciju hijerarhije podataka, ali umanjuje učinkovitost pregleda te je zapravo korak unatrag prema relacijskom modelu. Pokazano je da se normalizacijom dimenzijskih tablica uštedi jako malo prostora pa mnogi smatraju da se ovakav model nikada ne treba koristiti [7]. Slika 8 prikazuje primjer skladišta podataka prema pahuljastom modelu.



Slika 8. Pahuljasti model skladišta podataka [13]

2.7. Usporedba transakcijske baze podataka i skladišta podataka

Transakcijske baze podataka i skladišta podataka se uvelike razlikuju, ali nemoguće je pričati o jednome bez drugoga. Toj tezi ide u prilog i činjenica da je skladište podataka, kao ideja, nastalo kao pomoć pri korisničkim zahtjevima koje transakcijska baza nije mogla uspješno izvršiti, a također skladište podataka kao izvor najčešće koristi jednu ili više transakcijskih baza. S obzirom da većina IT stručnjaka, a posebice administratora baza podataka, svoju karijeru počinje na transakcijskim bazama te po potrebi prijeđe na rad nad skladištima podataka, razumijevanje razlike između ova dva pojma je ključno i veoma korisno. Najvažnije razlike između transakcijske baze i skladišta podataka sažete su u tablici 1.

Tablica 1. Razlike između transakcijskih sustava i skladišta podataka

Transakcijski sustav	Skladište podataka
Sadrži trenutne podatke	Sadrži povijesne podatke
Pohranjuje detaljne podatke	Pohranjuje detaljne i sumarne podatke
Podaci su promjenjivi	Podaci su postojani
Velika učestalost transakcija	Srednja ili mala učestalost transakcija
Predvidljivi načini korištenja (ponavljaju se)	Nepredvidljivi načini korištenja
Orijentiran ka dnevnim operacijama i vođenju poslovnog sustava	Orijentiran ka analizi podataka
Potpora dnevnim, operativnim odlukama	Potpora strateškim odlukama
Poslužuje velik broj operativnih korisnika	Poslužuje manji broj korisnika obično pozicioniranih u upravljačkim strukturama poduzeća (mada postoji trend sve veće dostupnosti skladišta podataka svim članovima poduzeća kao potpora svih vrsta odluka)
Izuzetno važna raspoloživost	Manje važna raspoloživost
Težište na pohranjivanju podataka	Težište na dobavljanju informacija

3. OLAP SUSTAV

Online analitičko procesiranje (engl. *online analytical processing, OLAP*) je odgovor na višedimenzionalne upite nekog sustava. OLAP je dio šireg pojma poslovne inteligencije koji obuhvaća i relacijske baze, izrade izvješća te analize podataka. Tipično korištenje OLAP-a uključuje poslovne izvještaje za prodaju, marketing, poslovno upravljanje procesa, budžetiranje i predviđanje, financijske izvještaje itd. [14]. U nastavku ovog rada, opisat će se i usporediti dva najpoznatija modela OLAP sustava (višedimenzionalni model i tablični model) te njihove najvažnije karakteristike. Ovi modeli su svojstveni za alat Microsoft SSAS koji se koristi za analitičku obradu podataka u programu Microsoft SQL Server. Microsoft SSAS alat će se koristiti i detaljnije opisati u praktičnom dijelu ovog rada.

3.1. Višedimenzionalni modeli

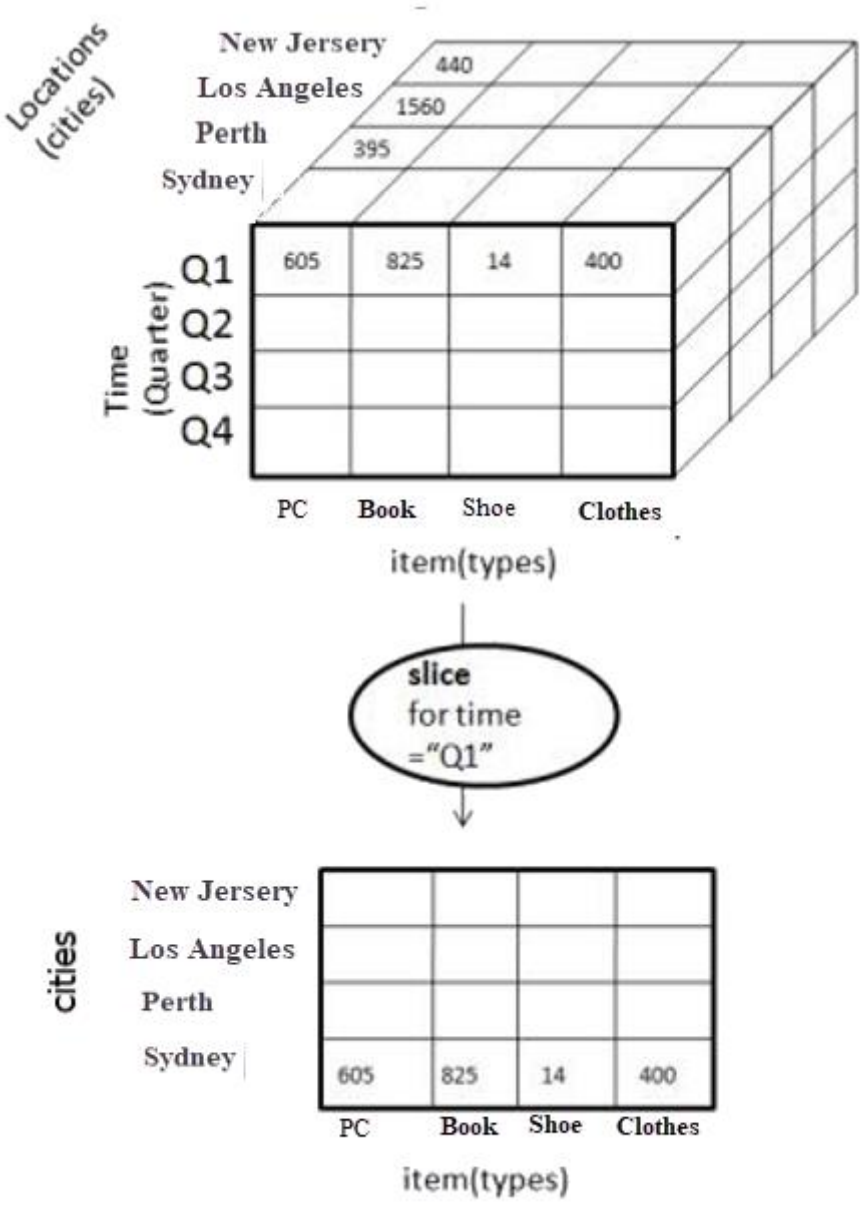
Višedimenzionalni model (tzv. OLAP kocka) je struktura podataka takva da dopušta brzu analizu s obzirom na prethodno objašnjene dimenzijske tablice koje definiraju poslovni problem. Višedimenzionalni model prevladava ograničenja relacijskih baza koja onemogućuju analizu i prikaz velikih grupa podataka. Iako postoje mnogi alati za izradu izvještaja iz relacijskih baza, ti alati su spori kada se radi nad cijelom relacijskom bazom. Problemi nastaju kada je potrebno napraviti izvještaj ili analizu iz drugih, višedimenzionalnih perspektiva. Korištenjem kocke korisnik dobiva brzu i praktički istovremenu interakciju sa podacima. Kocka se može promatrati kao nadogradnja na dvodimenzionalnu strukturu proračunskih tablica (engl. *sheets*) [15].

3.1.1. Vrste operacija nad višedimenzionalnim modelom

U ovom poglavlju opisat će se glavne operacije nad kockom kako bi se što bolje objasnila višedimenzionalna priroda skladišta podataka. Russo i Ferrari, u svojoj knjizi naziva „*Tabular modeling in Microsoft SQL server analysis services*“ govore o pet operacija koje su opisane u nastavku [15]. Primjer za svaku operaciju je fiktivno poduzeće koje se bavi prodajom knjiga, obuće i odjeće te računala. Prodaja se promatra nad jednom poslovnom godinom podijeljenom na kvartale u gradovima Perth, Sydney, Los Angeles i New Jersey [16].

a) Slice (hrv. rezanje)

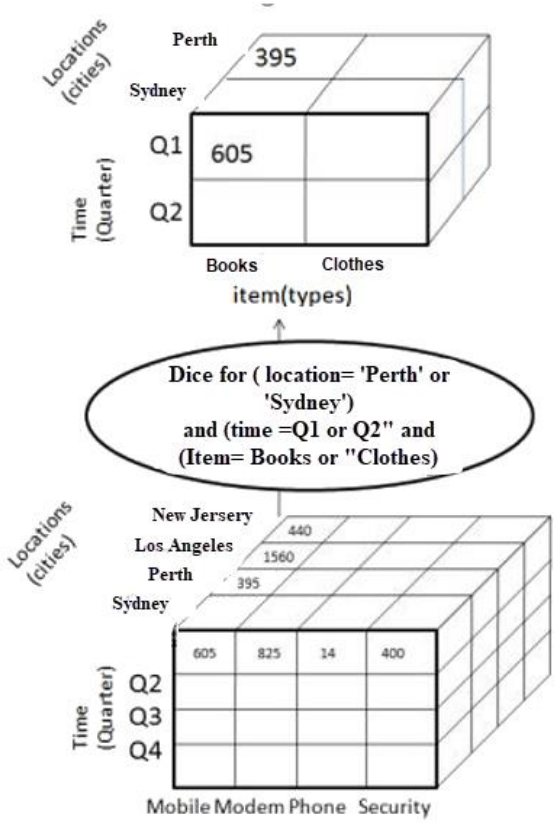
Slice je operacija uzimanja pravokutnog podskupa kocke birajući jednu vrijednost za neku dimenzijsku tablicu, pritom stvarajući kocku sa jednom dimenzijom manje [15]. Slika 9 prikazuje *Slice* operaciju prilikom koje se reže po vremenskoj dimenziji te se prikazuju podaci prodaje za prvi kvartal u godini [16].



Slika 9. Prikaz *Slice* operacije [16]

b) Dice (hrv. kockica)

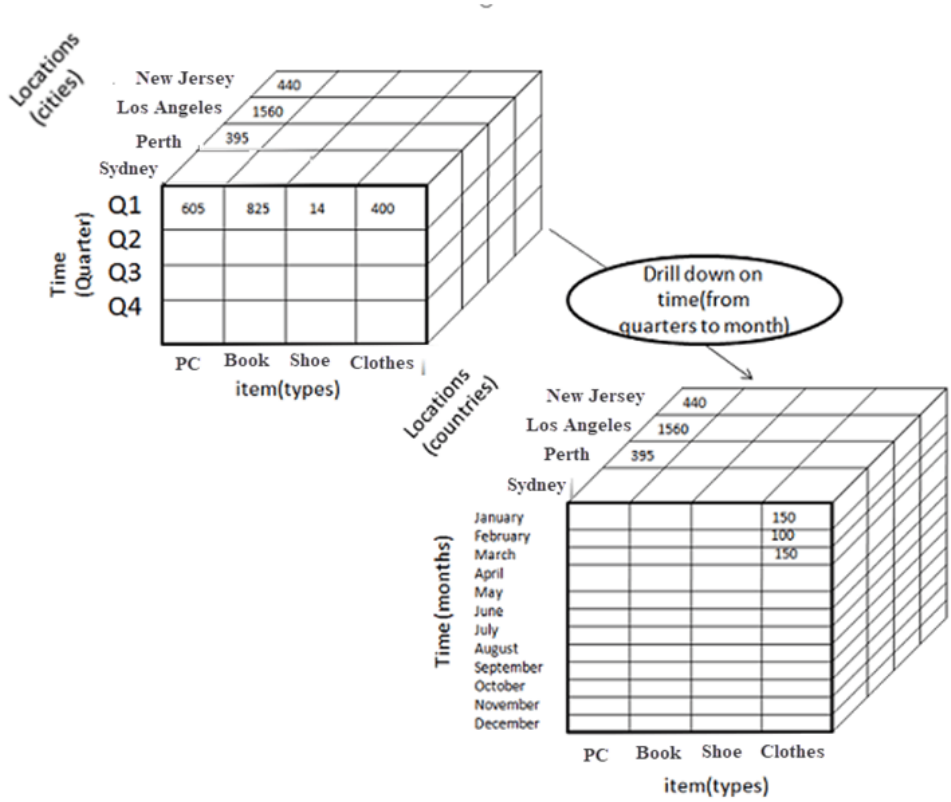
Dice operacija pravi podkocku, tj. manju kocku dopuštajući analitičaru da bira specifične vrijednosti iz više dimenzijskih tablica [15]. Slika 10 prikazuje *Dice* operaciju pri kojoj se uzima podkocka (tj. manja kocka) te se od ukupne kocke uzimaju u obzir podaci samo za gradove Perth i Sydney. Također se ovdje uzimaju u obzir rezultati iz prva dva kvartala za prodaju knjiga i odjeće [16].



Slika 10. Prikaz *Dice* operacije [16]

c) Drill (hrv. bušenje)

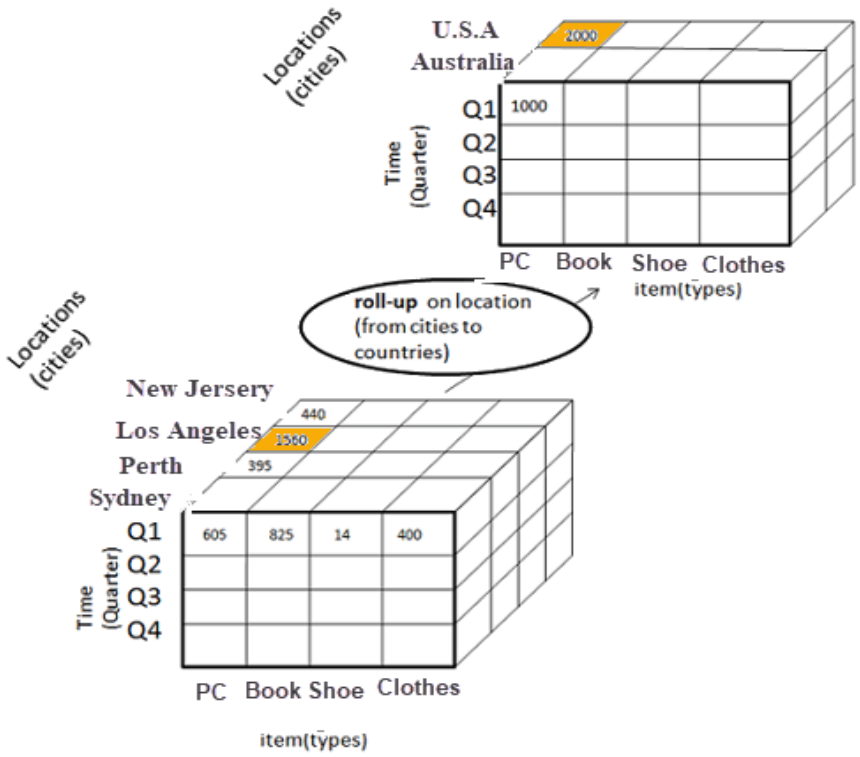
Drill operacija je vrsta operacije koja omogućuje korisniku pretraživanje razina podataka od sažetih do detaljnijih [15]. Slika 11 prikazuje izgled *Drill* operacije na primjeru vremenske dimenzije, tj. prikazuje detaljnije vremenski atribut podataka. Ovdje se više ne uzimaju kvartali kao vremenska dimenzija, već se ulazi u detalje odnosno u prikaz podataka po mjesecima u godini [16].



Slika 11. Prikaz *Drill* operacije [16]

d) Roll-up (hrv. sažimanje)

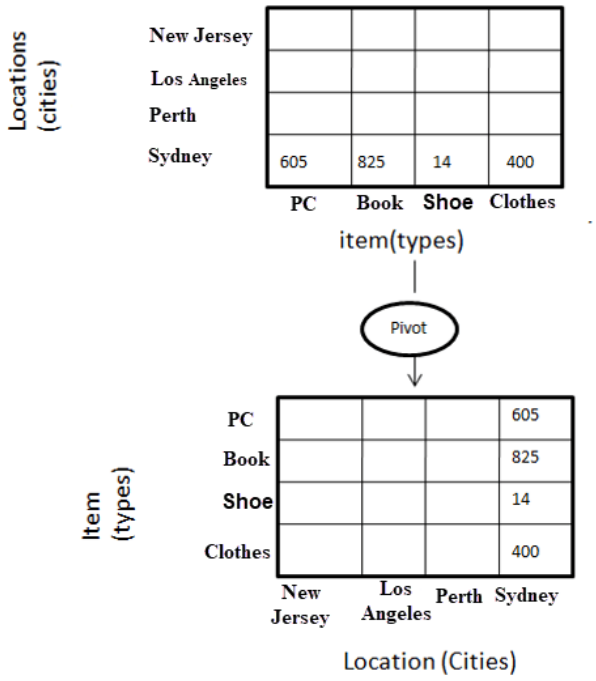
Roll-up operacija je zapravo suprotna *Drill* operaciji. Kod *Roll-up* operacije grupiraju se podaci unutar dimenzije po nekom zajedničkom nazivniku. Pravilo sažimanja može biti agregatna funkcija kao, primjerice, računanje profita kao razlika između prihoda i rashoda [15]. Slika 12 prikazuje grupiranje podataka na temelju lokacije tj. grupiranje gradova u njihove države. Ovom operacijom podaci se mogu gledati na višoj razini [16].



Slika 12. Prikaz *Roll-up* operacije [16]

e) Pivot (hrv. pivot)

Pivot operacija omogućuje rotaciju podataka tj. kocke kroz promatrane dimenzije. Primjerice vertikalna dimenzija se rotira u horizontalnu, a tu rotaciju također prate i podaci [15]. Slika 13 prikazuje primjer *Pivot* operaciju i rotaciju dvije promatrane dimenzije (tj. dimenziju lokacije i prodajnih artikala) [16].



Slika 13. Prikaz *Pivot* operacije [16]

3.2. Tablični model

Najveći mogući objekt kod tabličnog modela naziva se baza. Tablični model, kao koncept, temelji se na tablicama pri čemu se dodatno naglašava sličnost između tabličnog modela i relacijske baze podataka. Tablica u tabličnim modelima je najčešće jednaka jednoj tablici ili je rezultat upita nad više tablica u relacijskoj bazi podataka. Upiti prema tabličnim modelima se mogu izvoditi na dva načina. Prvi način svojstven za SSAS tablični model, koji će se koristiti u praktičnom dijelu rada, je *vertipaq* prilikom kojeg se upit obavlja nad podacima u RAM-u (engl. *random access memory*). RAM je oblik računalne memorije čijem se sadržaju može izravno pristupiti. Nadalje, kod *vertipaq-a* podaci su podijeljeni u odvojene stupčane strukture. Takva struktura omogućuje brz dohvat pojedinog stupca, ali činjenica je da su također mogući i problemi prilikom dohvata pojedinog retka unutar tablice sa svim njezinim stupcima. Budući da je potrebno cjelovito logičko skeniranje svakog stupca za svaki upit, podaci su često komprimirani u memoriju.

Drugi način upita prema tabličnom modelu je tzv. direktni upit (engl. *direct query*). Glavna prednost direktnog načina upita je garancija da su rezultat upita uvijek najažurniji podaci. Direktni način upita ne sprema podatke u RAM, već pristupa samo onim tablicama i stupcima potrebnim za izvršenje upita čime se dobije mogućnost slanja upita na veće baze nego što je

dostupno RAM-a. Također, preporučeno je koristiti direktni način upita za manje baze koje se često ažuriraju. Način rada direktnog upita se bazira na prevođenju iz programskih jezika svojstvenih tabličnom modelu u programski jezik za obavljanje upita nad relacijskom bazom podataka. Za kvalitetan rad direktnog upita potrebno je imati dobre performanse i optimiziranu relacijsku bazu podataka na koju se tablični model oslanja [15].

3.3. Usporedba modela

Višedimenzionalni modeli se sastoje od kocaka, dimenzijskih i činjeničnih tablica dok se tablični model sastoji od tablica i stupaca. S obzirom na to da su tablični modeli spremljeni u RAM, a višedimenzionalni na fizičkom disku, hardware je potpuno različit. Za tablični model bitno je imati što više memorije te bolju brzinu procesora. Ako je baza velika, tj. veća od 5 terabajta, tablični model se ne može implementirati. Višedimenzionalni modeli se oslanjaju na hard disk [17].

Implementiranjem tabličnog modela moguće je preskočiti kreiranje skladišta podataka i raditi modeliranje direktno iz ETL (engl. *extract, transform, load*) procesa, tj. procesa prilagodbe i integracije podataka. Nadalje, struktura višedimenzionalnih podataka je vrlo rigidna te ju je najbolje staviti u zvjezdastu shemu obzirom na to da je velika i kompleksna. U tabličnoj strukturi, model je relacijski te je ova struktura manja i samim time jednostavnija za korištenje [18].

Prednosti tabličnog modela su bolje performanse od višedimenzionalnog modela te u većini slučajeva i jednostavnije razvijanje. Prilikom izgradnje tabličnog modela vrši se kompresija podataka na jednu desetinu originalne baze što je vrlo bitno kada se model izgrađuje nad bazama preko terabajta veličine. Također, kao prednost ističe se mogućnost korištenja određenih operacija kao što je DISTINCT funkcija koja iz skupa podataka vraća različite podatke [18].

Prednost višedimenzionalnog modela nad tabličnim modelom je prvenstveno dugotrajnost i mogućnost rada sa velikim bazama podataka (preko 5 terabajta) te se ovaj model odlično nosi sa složenim zahtjevima. Dugotrajnost postojanja višedimenzionalnog modela je njegova najveća prednost te je moguće na internetu pronaći mnoštvo materijala i primjera na temu ovog modela. Također prednost višedimenzionalnog modela je u tome da korisnici, koji koriste ovaj model i zadovoljni su njegovim funkcionalnostima, neće se lako odlučiti za prelazak na tablični model s obzirom da migracija između ova dva modela nije moguća. Prelazak na tablični model iziskuje ponovnu izradu modela od samog početka, a taj proces može biti veoma skup i dugotrajan [17].

3.4. ETL proces

ETL je kratica koja se odnosi na dohvaćanje, transformaciju i učitavanje podataka, te čini glavni i najvažniji dio same izgradnje jednog skladišta podataka. ETL podrazumijeva proces dohvaćanja/preuzimanja podataka iz njihovih izvorišnih sustava i spremanje tih istih podataka u skladište podataka. U literaturi se navodi da je za ETL proces potrebno izdvojiti približno 70% vremena i resursa kada govorimo o izgradnji skladišta podataka [19].

ETL je integracijska funkcija koja uključuje ekstrakciju podataka iz vanjskih izvora, transformaciju tih podataka tako da su što bolje prilagođeni poslovnim potrebama i učitavanje tih podataka u skladište podataka. Podaci koji ulaze u ETL mogu doći iz različitih izvora kao što je transakcijska baza, Excel dokument , CRM alat, ERP aplikacija itd. [17].

Extract, Transform i Load su tri glavne funkcije ETL procesa i ove glavne značajke su opisane u nastavku [20].

1. *Extract* (hrv. *dohvaćanje*) je proces „čitanja“ podataka iz određenog izvora i ekstrakcije (uzimanja) željenog seta odnosno niza podataka. U ovom koraku, podaci se izvlače iz izvora u međuprostor skladišta (engl. *staging area*). Sve potrebne transformacije, odnosno prilagodbe podataka, vrše se u međuprostoru skladišta. Međuprostor skladišta daje mogućnost da se podaci provjere i validiraju prije nego ti isti podaci uđu u skladište podataka.
Tri najčešće korištene metode ekstrakcije odnosno izvlačenja podataka su:
a) Potpuni dohvat/izvlačenje (engl. *full extraction*)
b) Djelomični dohvat/izvlačenje (engl. *partial extraction*) bez obavijesti o ažuriranju
c) Djelomična ekstrakcija/izvlačenje (engl. *partial extraction*) sa obavijesti o ažuriranju
Tijekom izvlačenja podataka vrše se i određene provjere valjanosti kao što su:
a) usklađivanje zapisa sa izvornim podacima
b) provjera da nisu neki neželjeni podaci (engl. *spam*) učitani
c) provjera vrste podataka
d) uklanjanje svih duplikata

2. *Transform* (hrv. *transformacija*) je proces pretvaranja podataka iz jednog oblika u onaj koji se može učitati u skladište podataka. Ovaj proces uključuje i čišćenje podataka tako da ti podaci postanu dostupni za analizu. Dva pojma povezana s ovim procesom su čišćenje i konformiranje. Čišćenje se odnosi na ispravljanje pogrešnih podataka i dostavu ispravnih, čistih podataka krajnjim korisnicima. S druge strane, konformiranje podataka ima za cilj učiniti podatke točnima i kompatibilnima s ostalim matičnim podacima [21]. Transformaciji podataka može se pristupiti na dva načina [22][21]:
a) Klasičan pristup - podaci se dohvate, stave u međuprostor skladišta (tzv. pripremno skladište), prođu kroz potrebnu transformaciju i iza toga se učitaju u skladište podataka

b) ETL pristup – prvo se podaci dohvate i odmah stave odnosno učitaju u skladište podataka. Sve korekcije odnosno transformacije podataka se potom vrše u samom skladištu podataka.

Postoje dvije vrste transformacije podataka [23]:
a) Osnovne transformacije
1) Čišćenje se odnosi na osiguravanje dosljednosti podataka (*Male* postaje „M“, *Female* postaje „F“), konzistentnost u pisanju datuma itd.;
2) Uklanjanje duplikata se odnosi na identificiranje i uspješno uklanjanje, odnosno brisanje duplih podataka;
3) Revidiranje formata je pretvorba skupa znakova, pretvorba mjerne jedinice, pretvorba datuma/vremena;
4) Ključno restrukturiranje se odnosi na uspostavljanje ključnih odnosa među tablicama.

b) Napredne transformacije
1) Filtriranje - samo određeni retci ili stupci se odabiru;
2) Pridruživanje - povezivanje podataka iz više izvora;
3) Razdvajanje - razdvajanje jednog stupca u više stupaca;
4) Validiranje podataka - jednostavna ili složena provjera podataka;
5) Sažimanje - vrijednosti se sažimaju kako bi se dobili sumarni podaci.

3. *Load* (hrv. *učitavanje*) je proces učitavanja podataka odnosno zapisivanja podataka u ciljanu bazu podataka. U tipično skladište podataka, velike količine podataka moraju biti učitane u relativno kratkom vremenskom periodu, stoga sam proces učitavanja mora biti optimiziran [22]. U slučaju pogreške prilikom učitavanja podataka, mehanizmi za oporavak bi se trebali konfigurirati te se proces ponovno pokrenuti od trenutka pogreške kako bi se sačuvao integritet podataka. Administratori baze podataka bi trebali pratiti, puštati i otkazivati procese imajući u vidu performansu servera. Kada je riječ o učitavanju podataka, razlikujemo tri vrste učitavanja [24].
a) Početno učitavanje/punjenje (engl. *initial load*) – punjenje svih tablica u skladištu podataka
b) Postupno učitavanje/punjenje (engl. *incremental load*) – periodično učitavanje podataka; datum zadnje izmjene podataka se bilježi i nakon toga se učitavaju u skladište samo oni podaci poslije tog zadnjeg zabilježenog datuma.
c) Potpuno osvježavanje (engl. *full refresh*) – brisanje sadržaja iz jedne ili više tablica i ponovno učitavanje „svježih“ podataka.

3.5. Uloge i odgovornosti ETL stručnjaka

Obzirom na važnost ETL procesa, potrebno je naglasiti koje uloge i odgovornosti ETL stručnjak mora imati [20]:

- ETL stručnjak mora paziti na potrebe i zahtjeve organizacije (engl. *end users*). On/ona mora usko surađivati s više suradnika kako bi se razvila najbolja metoda i infrastruktura za uspješan ETL rad
- ETL stručnjak mora imati u vidu sve podatke i programe te nadgledati svaku ETL komponentu i njihove podkomponente
- ETL stručnjak je često središnja točka za razumijevanje različitih tehničkih standarda koji se trebaju razviti. On/ona mora također osigurati da je ETL proces ponavljajući, dokumentiran i mora paziti na promjene
- ETL stručnjak mora osigurati da su razni softverski alati, koji su potrebni da naprave različite vrste obrada podataka, pravilno odabrani.

3.6. Problemi kod ETL procesa

Postoje mnogi problemi odnosno izazovi kod implementiranja efikasnih i pouzdanih ETL procesa. Neki od često spomenutih problema su [20]:

- Tehničke poteškoće prilikom unošenja, integracije i transformacije podataka iz različitih heterogenih izvora
- Kratki vremenski okviri za učitavanje ili predugo učitavanje podataka
- Nedosljedna poslovna pravila teška za održavanje
- Nedostatak važnih podataka u izvorišnim sustavima
- Slabe performanse upita

Ukoliko izvorišni podaci nisu pročišćeni, pravilno izvedeni, transformirani i integrirani na pravilan način, ETL proces koji je „kralježnica“ skladišta podataka, ne može se obaviti.

4. IMPLEMENTACIJA SKLADIŠTA PODATAKA

U ovom dijelu rada demonstrirat će se i implementirati skladište podataka na primjeru fiktivne trgovine biciklima i biciklističkim dijelovima. Pokazat će se razvoj i implementacija skladišta podataka od odabira podataka i tablica, punjenje istih ETL procesom iz više heterogenih izvora te naposljetku izgradnja višedimenzionalnog modela. Radi jednostavnosti, obje korištene baze *AdventureWorks2019* baza podataka i *AdventureWorksDW2019* skladište podataka će biti na istom serveru i koristit će se niže navedene Microsoftove tehnologije i platforme.

4.1. Namjena i proces implementacije skladišta podataka

U ovom dijelu rada demonstrirat ću rad sa skladištem podataka na primjeru fiktivnog poduzeća. Prikazat ću implementaciju skladišta podataka koje prati prodaju dijelova za bicikle po prodajnom predstavniku u određenom vremenskom okviru.

Prvi korak koji ću obaviti je prepoznati potrebne podatke za implementaciju skladišta podataka. Najbolji način za to je zapravo napisati upit u relacijskoj bazi podataka koji kasnije želim postaviti nad skladištem podataka. Potrebno je također odlučiti do koje zrnatosti će se ići i koji podaci će biti potrebni korisniku i poslovnoj analizi. Nakon što se prepoznaju potrebni tj. zanimljivi podaci prikazat ću izradu tablica unutar skladišta podataka sa prikladnim nazivima tablica i stupaca unutar njih.

Sljedeći korak koji ću prikazati je kreiranje međuprostornih (engl. *staging*) tablica u koje će se spremati podaci iz ETL procesa te koje će biti slične strukturi tablica u relacijskoj bazi podataka. Nadalje, kreirat ću pogled unutar skladišta podataka koji će „gledati“ na podatke unutar relacijskih baza podataka. Napisat ću međuprostornu proceduru koja će *merge* operacijom popuniti podatke unutar međuprostornih tablica. Nakon unosa podataka u međuprostornu tablicu napisat ću i prikazati proceduru koja će obavljati transformacije nad podacima unutar međuprostorne tablice te će se *merge* operacijom podaci dodati u dimenzijske i činjenične tablice.

Također, prikazat ću i ETL proces iz različitih izvora pored relacijske baze podataka. U SSIS-u ću prikazati prijenos podataka iz Excel datoteke, tekstualne datoteke te XML datoteke. Model prikazanog skladišta podataka će biti zvjezdasti tj. neće postojati veza između dviju dimenzijskih tablica već će sve dimenzijske tablice imati vezu na činjeničnu tablicu. Dimenzijske tablice će se puniti iz relacijske baze podataka, dok će se vremenska dimenzija i činjenična tablica puniti iz ostalih heterogenih izvora koji će biti opisani u nastavku rada.

4.2. Korištene tehnologije

Slijedi kratki opis tehnologija i platformi koje će se koristiti u ovom dijelu rada [25].
a) SSMS (*SQL Server Management Studio*) - *SQL Server Management Studio* je softverska aplikacija koja je prvi put pokrenuta s Microsoft SQL Serverom 2005 i koristi se za konfiguriranje i upravljanje svim komponentama unutar Microsoft SQL Servera.
b) SSIS (*SQL Server Integration Services*) - *SQL Server Integration Services* komponenta je softvera baze podataka Microsoft SQL Server koja se može koristiti za obavljanje širokog spektra zadataka vezanih uz migracije podataka. SSIS je platforma za integraciju podataka i aplikacija za radni tijek.
c) SSAS (*SQL Server Analysis Services*) - *Microsoft SQL Server Analysis Services* je mrežni alat za analitičku obradu i vađenje podataka u programu Microsoft SQL Server. SSAS se koristi u organizacijama kao alat za analizu i dodjeljivanje smisla informacijama koje su možda rasprostranjene u više baza podataka ili u različitim tablicama odnosno datotekama.
d) *Visual Studio* - *Microsoft Visual Studio* integrirano je razvojno okruženje tvrtke Microsoft. Koristi se za razvoj računalnih programa kao i *web* stranica, *web* aplikacija, *web* usluga i mobilnih aplikacija.
e) Integrirano razvojno okruženje (engl. *integrated development environment*, IDE) – Integrirano razvojno okruženje softverski je program koji računalnim programerima pruža sveobuhvatne pogodnosti za razvoj softvera. IDE se obično sastoji od uređivača izvornog koda, alata za automatsku izgradnju projekta (engl. *build automation*) i alata za uklanjanje pogrešaka.
f) AdventureWorks2019 baza podataka i AdventureWorksDW2019 skladište podataka su Microsoftove testne baze podataka koje sadrže podatke za fiktivno multinacionalno poduzeće Adventure Works Cycles koje se bavi proizvodnjom i prodajom bicikala i biciklističkih dijelova. Putem ovih baza Microsoft želi pokazati sve opcije SQL servera te se često Microsoftova dokumentacija, brojne knjige i primjeri programerskog koda temelje upravo na njima. Obje baze podataka su dostupne *online* [26].
g) SQL (engl. *Structured Query Language*) – SQL je strukturirani upitni jezik tj. programski jezik visoke razine. Najpopularniji je računalni jezik za izradu, traženje, ažuriranje i brisanje podataka iz relacijskih baza podataka. SQL je standardiziran preko standarda ANSI (engl. *American National Standards Institute*) i ISO standarda (engl. *International Organization for Standardization*).
h) Kutija za alat (engl. *toolbox*) – Ovaj pojam se odnosi na set softverskih alata, jednostavno dostupnim iz jednog izbornika.
i) PowerBI – Power BI je usluga poslovne analitike tvrtke Microsoft. Cilj mu je pružiti interaktivne vizualizacije i mogućnosti poslovne inteligencije s dovoljno jednostavnim sučeljem da krajnji korisnici mogu stvoriti vlastita izvješća i nadzorne ploče. Dio je Microsoft Power Platforme.

4.3. Definicije osnovnih pojmova

Nadalje, za što bolje shvaćanje ovog dijela rada, neophodno je definirati i objasniti osnovne pojmove potrebne za razumijevanje implementacije skladišta podataka. Opis važnih pojmova slijedi niže [27].

a) Pogled (engl. *view*) – ovaj pojam se odnosi na virtualnu tablicu baziranu nad skupom podataka koji je rezultat upita nad više tablica. Kao i tablica, *view* se sastoji od imenovanih stupaca i podataka. *View* dopušta spajanje više tablica u jedinstven skup podataka koji se kasnije može koristiti kao jedinstvena tablica unutar drugih procesa.

b) Procedura (engl. *stored procedure*) – procedura je programski kod koji se može sačuvati i koristiti neograničeno mnogo puta. Koristi se za programski kod koji je potrebno često izvoditi. Procedura može primiti ulazne varijable te obavljati operacije ovisno o varijabli.

c) Operacija dodavanja (engl. *insert*) – operacija dodavanja podataka u bazu podataka.

d) Operacija ažuriranja (engl. *update*) – operacija ažuriranja podataka u bazi podataka.

e) Operacija brisanja (engl. *delete*) – operacija brisanja podataka u bazi podataka.

f) Operacija sjedinjenja/spajanja (engl. *merge*) – operacija u SQL programskom jeziku koja u jednom potezu može obavljati sve tri glavne operacije nad podacima: ažuriranje, brisanje i dodavanje podataka. *Merge* operacija uspoređuje dvije tablice po odabranom ključnom stupcu tj. prema podacima unutar stupca, te je moguće, ovisno o tome jesu li podaci isti u oba stupca, obaviti jednu od gore navedene tri operacije.

g) Primarni ključ (engl. *primary key*) - u relacijskom modelu baza podataka, primarni ključ je specifičan izbor minimalnog skupa atributa koji jedinstveno identificira subjekt. Neformalno, primarni ključ je odgovor na pitanje: "Koji atributi jednoznačno identificiraju zapis ?", a u jednostavnim je slučajevima primarni ključ jedan atribut: jedinstveni ID.

h) Strani ključ (engl. *foreign key*) - strani ključ je skup atributa u tablici koji se odnosi na primarni ključ druge tablice te ne mora imati jedinstvenu vrijednost. Strani ključ opisuje odnos bilo koje dvije tablice u bazi podataka.

i) Naseljavanje (engl. *deployment*) – ovaj pojam uključuje sve korake, procese i aktivnosti koji su potrebni da se napravi jedan softverski sustav ili ažurira postojeći sustav krajnjim korisnicima. Danas, većina IT poduzeća i programera, proces naseljavanja obavlja kombinacijom ručnih i automatskih procesa. Neke od najčešćih aktivnosti naseljavanja su puštanje softvera (engl. *software release*), instaliranje, testiranje, naseljavanje i nadziranje performansi (engl. *performance monitoring*).

4.4. Odabir podataka iz relacijske baze

Zanima nas prodaja dijelova za bicikle po prodajnom predstavniku u određenom vremenskom okviru u gore navedenom fiktivnom poduzeću Adventure Works Cycles. Podaci koji nas zanimaju su informacije o proizvodima, zaposlenicima i prodaji tj. zanima nas koji proizvod se prodaje te u kojoj količini po pojedinom prodajnom predstavniku. Kako bi se došlo do tih informacija, potrebno je pronaći i ispisati upit u relacijskoj bazi podataka. Slika 14 prikazuje korišteni upit (engl. *query*). Ovako formulirani upit dohvaća sve podatke o proizvodima, prodaji i prodajnim predstavnicima poduzeća koje se za ovu svrhu analizira.

```
SELECT *
FROM Production.Product AS p
INNER JOIN Sales.SalesOrderDetail AS aod ON aod.ProductID = p.ProductID
INNER JOIN Sales.SalesOrderHeader AS aoh ON aoh.SalesOrderID = aod.SalesOrderID
INNER JOIN Sales.SalesPerson AS ap ON ap.BusinessEntityID = aoh.SalesPersonID
INNER JOIN HumanResources.Employee AS e ON e.BusinessEntityID = ap.BusinessEntityID
INNER JOIN Production.ProductModel AS pm ON pm.ProductModelID = p.ProductModelID
INNER JOIN Production.ProductSubcategory AS ps ON ps.ProductSubcategoryID = p.ProductSubcategoryID
INNER JOIN Production.ProductCategory AS pc ON pc.ProductCategoryID = ps.ProductCategoryID
INNER JOIN Person.Person AS per ON per.BusinessEntityID = e.BusinessEntityID
INNER JOIN Person.PersonPhone AS pp ON pp.BusinessEntityID = per.BusinessEntityID
INNER JOIN Person.EmailAddress AS ea on ea.BusinessEntityID = per.BusinessEntityID
INNER JOIN person.BusinessEntityAddress AS bea ON bea.BusinessEntityID = ap.BusinessEntityID
INNER JOIN person.Address AS a ON a.AddressID = bea.AddressID
```

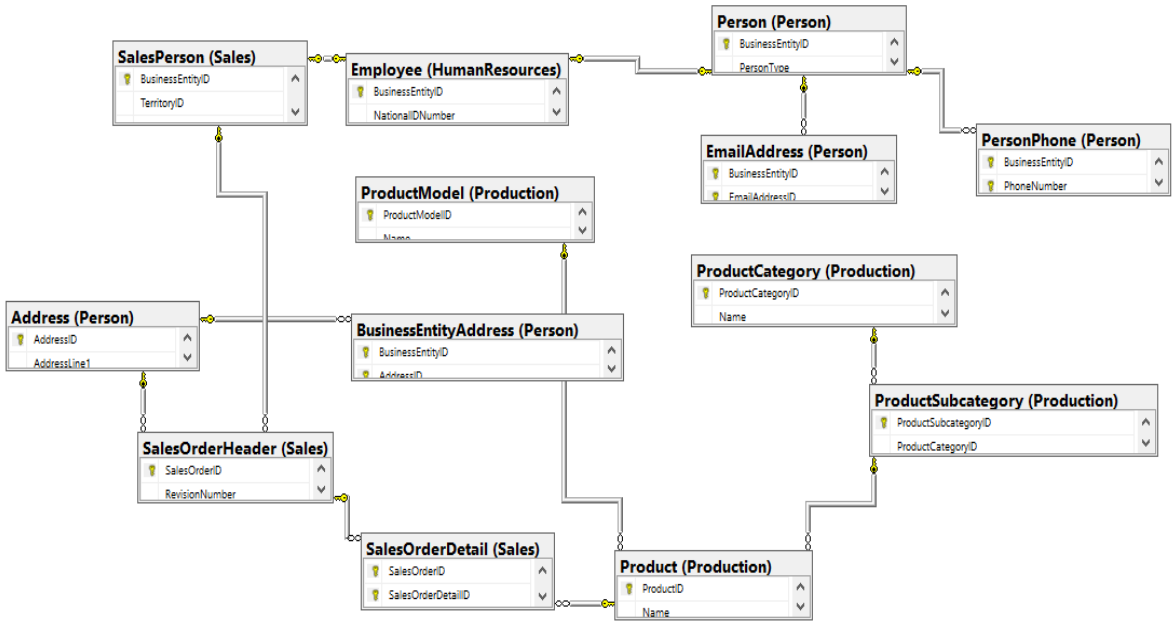
Slika 14. Prikaz upita u relacijskoj bazi podataka

Za bolje razumijevanje upita, potrebno je istaknuti koje podatke sadrži pojedina tablica iz upita. Slijedi kratki opis tablica.

- Product (hrv. *proizvod*) – sadrži sve informacije o proizvodima ovog poduzeća.
- SalesOrderDetail (hrv. *detalji prodaje*) – sadrži detaljnije informacije o pojedinoj prodaji tj. računu po pojedinom proizvodu odnosno stavci računa.
- SalesOrderHeader (hrv. *zaglavlje računa*) – sadrži glavne i najvažnije informacije koje sačinjavaju jedan račun.
- SalesPerson (hrv. *prodajni predstavnik*) – sadrži prodajni učinak i ostale poslovne informacije za pojedinog prodajnog predstavnika.
- Employee (hrv. *zaposlenik*) – sadrži detaljnije osobne podatke o pojedinom zaposleniku ovog poduzeća.
- ProductModel (hrv. *model proizvoda*) – sadrži informacije o svim modelima proizvoda koje prodaje ovo poduzeće.
- ProductSubcategory (hrv. *potkategorija proizvoda*) – sadrži informacije o potkategorijama proizvoda.
- ProductCategory (hrv. *kategorija proizvoda*) – sadrži informacije o kategorijama proizvoda.

- Person (hrv. *osoba*) – sadrži informacije tj. ime i prezime osoba povezanih s ovim poduzećem. Tu se prikazuju imena klijenata, ali i imena zaposlenika koje je moguće iščitati samo iz ove tablice.
- PersonPhone (hrv. *telefonski broj osobe*) – sadrži informacije o telefonskim brojevima klijenata i zaposlenika.
- EmailAddress (hrv. *email adresa*) – sadrži informacije o email adresi klijenata i zaposlenika.
- BusinessEntityAddress (hrv. *poslovna adresa*) – sadrži strane ključeve tj. veze na adrese i tipove adresa.
- Address (hrv. *adresa*) – sadrži adrese svih klijenata i zaposlenika.

Slika 15 prikazuje dijagram dijela relacijske baze podataka AdventureWorks2019 koji prikazuje odnosno sadrži sve tablice i veze među njima iz prethodno opisanog upita.



Slika 15. Dijagram relacijske baze podataka

Sljedeći korak u ovom procesu je implementacija navedenih tablica u skladište podataka. Obzirom da je razina zrnatosti relacijske baze podataka najveće razine te za skladište podataka nije potrebno ići toliko u detalje pojedinih atributa, prikazat će se grupirani podaci iz više tablica relacijske baze podataka u jednoj dimenzijskoj tablici unutar skladišta podataka.

4.5. Izgradnja skladišta podataka

Izgradnja skladišta podataka sadrži sljedeće korake: izrada dimenzijskih tablica, punjenje dimenzijskih tablica, izrada i punjenje vremenske dimenzije, izrada činjenične tablice te punjenje činjenične tablice podacima. U sljedećem dijelu rada svaki od ovih navedenih koraka će se opisati te će se, gdje je to moguće, slikom prikazati pojedini proces.

4.5.1. Izrada dimenzijskih tablica

Nakon odabira svih potrebnih podataka prikazanih u upitu, slijedi izrada dimenzijskih tablica. Dimenzijska tablica naziva „dimProduct“, koja predstavlja proizvod, sastojat će se od osnovnih podataka o svim proizvodima Adventure Works Cycles poduzeća. Podaci u naredbi za kreiranje tablice tj. podaci koji opisuju pojedini proizvod su: IDProduct (hrv. *redni broj proizvoda*), ProductName (hrv. naziv proizvoda), ProductNumber (hrv. broj proizvoda), ProductModelName (hrv. naziv modela proizvoda), ProductCategoryName (hrv. naziv kategorije proizvoda), ProductSubCategoryName (hrv. naziv potkategorije proizvoda). Slika 16 prikazuje korištenu naredbu za kreiranje dimenzijske tablice „dimProduct“.

```
CREATE TABLE [dbo].[DimProduct]
(
    IDProduct int IDENTITY(1, 1) NOT NULL
  , ProductName nvarchar(50) NOT NULL
  , ProductModelName nvarchar(50) NOT NULL
  , ProductSubcategoryName nvarchar(50) NOT NULL
  , ProductCategoryName nvarchar(50) NOT NULL
  , CONSTRAINT [PK_DimProduct_IDProduct] PRIMARY KEY CLUSTERED ([IDProduct] ASC)
) ON [PRIMARY];
```

Slika 16. Kreiranje tablice "DimProduct"

Dimenzijska tablica naziva „dimEmployee“, koja predstavlja zaposlenika, sastojat će se od osnovnih podataka o svim zaposlenicima ovog poduzeća. Podaci u naredbi za kreiranje tablice tj. podaci koji opisuju pojedinog zaposlenika su: IDEmployee (hrv. *primarni ključ zaposlenika*), FirstName (hrv. *ime zaposlenika*), LastName (hrv. *prezime zaposlenika*), Address (hrv. *adresa zaposlenika*), Email (hrv. *email zaposlenika*), i PhoneNumber (hrv. *broj mobitela zaposlenika*).

Slika 17 prikazuje korištenu naredbu za kreiranje dimenzijske tablice „dimEmployee“.

```
CREATE TABLE [dbo].[DimEmployee]
(
    IDEmployee int IDENTITY(1, 1) NOT NULL
  , FirstName nvarchar(50) NOT NULL
  , LastName nvarchar(50) NOT NULL
  , Address nvarchar(50) NOT NULL
  , Email nvarchar(50) NOT NULL
  , PhoneNumber nvarchar(50) NOT NULL
  CONSTRAINT [PK_DimEmployee_IDEmployee] PRIMARY KEY CLUSTERED ([IDEmployee]
);
```

Slika 17. Kreiranje tablice "DimEmployee"

4.5.2. Punjenje dimenzijskih tablica podacima

Sljedeći korak u ovom procesu je punjenje dimenzijskih tablica podacima iz relacijske baze podataka. Kako bi se dimenzijske tablice napunile podacima, potrebno je kreirati pogled nad podacima u relacijskoj bazi podataka. S obzirom da će se skladište podataka izgraditi s manjom zrnatošću od relacijske baze podataka, moraju se kreirati pogledi nad tablicama kako bi se podaci grupirali i kako bi se same dimenzijske tablice napunile podacima. Za ovu svrhu korištena su dva pogleda.

Prvi pogled promatra informacije o proizvodu koje se žele unijeti u skladište podataka prema izgledu tablice „dimProduct“. Naziv i broj proizvoda dohvaćaju se iz tablice „Product“, naziv modela proizvoda dohvaća se iz tablice „ProductModel“, naziv potkategorije proizvoda dohvaća se iz tablice „ProductSubcategory“ i naposljetku naziv kategorije proizvoda dohvaća se iz tablice „ProductCategory“. Slika 18 prikazuje korištenu naredbu za kreiranje pogleda „vw_Product“ nad podacima u relacijskoj bazi podataka.

```
CREATE VIEW dbo.vw_Product
AS
    SELECT
        p.Name AS ProductName
      , p.ProductNumber
      , pm.Name AS ProductModelName
      , ps.Name AS ProductSubcategoryName
      , pc.Name AS ProductCategoryName
    FROM AdventureWorks2019.Production.Product AS p
        INNER JOIN AdventureWorks2019.Production.ProductModel AS pm ON pm.ProductModelID = p.ProductModelID
        INNER JOIN AdventureWorks2019.Production.ProductSubcategory AS ps ON ps.ProductSubcategoryID = p.ProductSubcategoryID
        INNER JOIN AdventureWorks2019.Production.ProductCategory AS pc ON pc.ProductCategoryID = ps.ProductCategoryID;
```

Slika 18. Kreiranje pogleda "vw_Product" [28]

Punjenje dimenzijske tablice obavlja se procedurom. U proceduri „dimProductMerge“ uspoređuju se podaci o proizvodima između pogleda „vw_Product“ i dimenzijske tablice „DimProduct“ po stupcu „ProductName“ gdje pogled služi kao izvor podataka, a dimenzijska tablica kao odredište podataka. U slučaju da podatak o nazivu proizvoda postoji u pogledu, a ne postoji u dimenzijskoj tablici, taj podatak će se dodati u dimenzijsku tablicu kao rezultat ove usporedbe. Slika 19 prikazuje proceduru za punjenje „DimProduct“ tablice.

```
CREATE PROCEDURE dbo.MergeDimProduct
AS
BEGIN
MERGE dbo.DimProduct AS trg
USING ( SELECT
          ProductName
        , ProductNumber
        , ProductModelName
        , ProductSubcategoryName
        , ProductCategoryName
    FROM dbo.vw_Product
      ) src ON src.ProductName = trg.ProductName
WHEN NOT MATCHED BY TARGET
INSERT INTO dbo.DimProduct
(
    ProductName, ProductNumber, ProductModelName, ProductSubcategoryName, ProductCategoryName
)
SELECT
          src.ProductName
        , src.ProductNumber
        , src.ProductModelName
        , src.ProductSubcategoryName
        , src.ProductCategoryName;
END
```

Slika 19. Procedura za punjenje "DimProduct" tablice [28]

Sljedeći korak je punjenje tablice “DimEmployee”. Kao i u prethodnom opisanom koraku, proces punjenja počinje pogledom nad potrebnim podacima u relacijskoj bazi podataka. Ime i prezime zaposlenika dohvaćamo iz tablice „Person“, adresu zaposlenika dohvaćamo iz tablice „Address“, email iz tablice „EmailAddress“ i broj mobitela zaposlenika dohvaćamo iz tablice „PersonPhone“. Slika 20 prikazuje korištenu naredbu za kreiranje pogleda „vw_Employee“.

```
CREATE VIEW vw_Employee
AS
    SELECT
        p.FirstName
      , p.LastName
      , a.AddressLine1
      , ea.EmailAddress
      , pp.PhoneNumber
    FROM [AdventureWorks2019].[Sales].[SalesPerson] sp
        LEFT JOIN HumanResources.Employee AS e ON e.BusinessEntityID = sp.BusinessEntityID
        LEFT JOIN Person.Person AS p ON p.BusinessEntityID = e.BusinessEntityID
        LEFT JOIN Person.PersonPhone AS pp ON pp.BusinessEntityID = p.BusinessEntityID
        LEFT JOIN Person.EmailAddress AS ea ON ea.BusinessEntityID = p.BusinessEntityID
        LEFT JOIN Person.BusinessEntityAddress AS bea ON bea.BusinessEntityID = p.BusinessEntityID
        LEFT JOIN Person.Address AS a ON a.AddressID = bea.AddressID;
```

Slika 20. Kreiranje pogleda "vw_Employee" [28]

U proceduri „dimEmployeeMerge“ uspoređuju se podaci o zaposlenicima između pogleda „vw_Employee“ i dimenzijske tablice „dimEmployee“ po stupcima „FirstName“ i „LastName“ gdje pogled služi kao izvor podataka, a dimenzijska tablica kao odredište podataka. U slučaju da podatak o imenu i prezimenu zaposlenika postoji u pogledu, a ne postoji u dimenzijskoj tablici, taj podatak će se dodati u dimenzijsku tablicu kao rezultat usporedbe. Slika 21. prikazuje proceduru za punjenje „DimEmployee“ tablice.

```
CREATE PROCEDURE dbo.MergeDimEmployee
AS
BEGIN
MERGE dbo.DimEmployee AS trg
USING ( SELECT
          FirstName
        , LastName
        , AddressLine1
        , EmailAddress
        , PhoneNumber
    FROM dbo.vw_Employee
      ) src ON src.FirstName = trg.FirstName AND src.LastName = trg.LastName
WHEN NOT MATCHED BY TARGET
INSERT INTO dbo.DimEmployee (FirstName, LastName, Address, Email, PhoneNumber)
SELECT
          src.FirstName
        , src.LastName
        , src.AddressLine1
        , src.EmailAddress
        , src.PhoneNumber;
END
```

Slika 20 Procedura za punjenje "DimEmployee" tablice

[28]

4.5.3. Kreiranje i punjenje vremenske dimenzije

Vremenska dimenzija je najvažnija dimenzija skladišta podataka. Skladište podataka je povijesna baza podataka za izvještavanje i analizu te je logičan zaključak da je za svaki podatak potrebno znati vrijeme kreiranja tog podatka. Tablica „DimDate“ tj. vremenska dimenzija puni se samo jednom te će jedan redak u tablici označavati jedan dan. Za razliku od ostalih primarnih ključeva u bazi, primarni ključ „IDDate“ tablice „DimDate“ ne povećava se automatski pravilnim redoslijedom (npr. 1,2,3,..) kao ostale tablice već će primarni ključ biti izvedenica datuma u formatu „*yyyymmdd*“ gdje *yyyy* označava godinu, *mm* mjesec u godini te *dd* dan u godini. Podaci za vremensku dimenziju će se učitavati iz Excel datoteke. Neće se obavljati nikakve transformacije već će se vremenska dimenzija direktno učitati u tablicu „DimDate“. Budući da je vremenska dimenzija sastavni dio svakog skladišta podataka, već spremni podaci se lako mogu pronaći u bilo kojem dostupnom gotovom rješenju skladišta podataka.

Vremenska dimenzijska tablica će se puniti kroz SSIS komponente koje uključuju komponentu za rad sa Excel datotekama („Excel Source“) i komponentu odredišta podataka („OLE DB Destination“) u kojoj se odabire tablica unutar skladišta podataka u koju želimo spremiti podatke iz Excel datoteke. Excel datoteka „DimDate“ sadrži podatke vremenske dimenzije tj. detaljne podatke na razini jednog dana. Korištena „DimDate“ datoteka dostupna je *online* za preuzimanje i korištenje [28]**Error! Reference source not found.**. Nakon mapiranja stupaca iz Excel datoteke „DimDate“ na stupce iz tablice „DimDate“ u skladištu podataka AdventureWorksDW2019, podaci su spremni za prijenos u skladište podataka. Slika 21 prikazuje naredbu za kreiranje tablice „DimDate“.

```
CREATE TABLE [dbo].[DimDate]
(
    [DateKey] [int] NOT NULL
  , [Date] [date] NOT NULL
  , [Day] [tinyint] NOT NULL
  , [DaySuffix] [char](2) NOT NULL
  , [Weekday] [tinyint] NOT NULL
  , [WeekDayName] [varchar](10) NOT NULL
  , [WeekDayName_Short] [char](3) NOT NULL
  , [WeekDayName_FirstLetter] [char](1) NOT NULL
  , [DOWInMonth] [tinyint] NOT NULL
  , [DayOfYear] [smallint] NOT NULL
  , [WeekOfMonth] [tinyint] NOT NULL
  , [WeekOfYear] [tinyint] NOT NULL
  , [Month] [tinyint] NOT NULL
  , [MonthName] [varchar](10) NOT NULL
  , [MonthName_Short] [char](3) NOT NULL
  , [MonthName_FirstLetter] [char](1) NOT NULL
  , [Quarter] [tinyint] NOT NULL
  , [QuarterName] [varchar](6) NOT NULL
  , [Year] [int] NOT NULL
  , [YearMonth] [char](6) NOT NULL
  , [MMYYYY] [char](6) NOT NULL
  , [MonthYear] [char](7) NOT NULL
  , [IsWeekend] [bit] NOT NULL
  , [IsHoliday] [bit] NOT NULL
  , [HolidayName] [varchar](20) NULL
  , [SpecialDays] [varchar](20) NULL
  , [FinancialYear] [int] NULL
  , [FinancialQuater] [int] NULL
  , [FinancialMonth] [int] NULL
  , [FirstDateofYear] [date] NULL
  , [LastDateofYear] [date] NULL
  , [FirstDateofQuater] [date] NULL
  , [LastDateofQuater] [date] NULL
  , [FirstDateofMonth] [date] NULL
  , [LastDateofMonth] [date] NULL
  , [FirstDateofWeek] [date] NULL
  , [LastDateofWeek] [date] NULL
  , [CurrentYear] [smallint] NULL
  , [CurrentQuater] [smallint] NULL
  , [CurrentMonth] [smallint] NULL
  , [CurrentWeek] [smallint] NULL
  , [CurrentDay] [smallint] NULL
  , PRIMARY KEY CLUSTERED ([DateKey] ASC)
) ON [PRIMARY];
GO
```

Slika 21. Kreiranje tablice "DimDate"

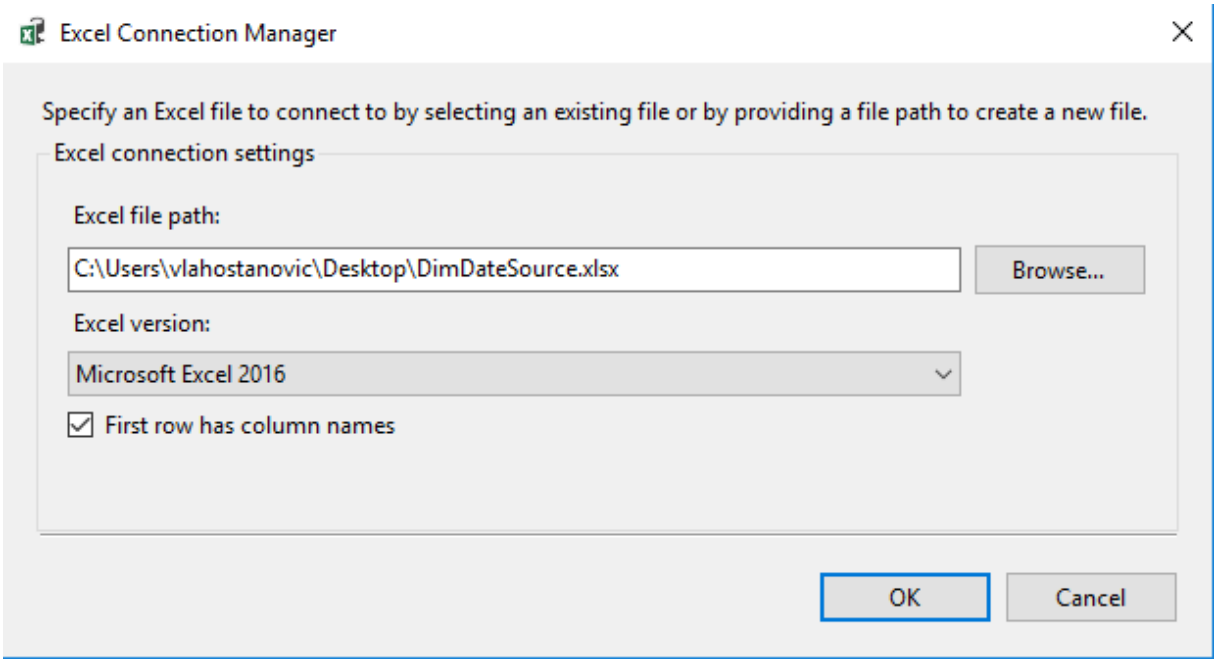
[28]

Slika 22 prikazuje vremenske podatke koji se nalaze u Excel datoteci „DimDate“.

IDDate	Date	Day	DaySuffix	Weekday	WeekDayName	WeekDayName_Short	WeekDayName_FirstLetter	DOWInMonth	DayOfYear	W
20210206	6.2.2021	6	th	7	Saturday	SAT	S	6	37	
20210207	7.2.2021	7	th	1	Sunday	SUN	S	7	38	
20210208	8.2.2021	8	th	2	Monday	MON	M	8	39	
20210209	9.2.2021	9	th	3	Tuesday	TUE	T	9	40	
20210210	10.2.2021	10	th	4	Wednesday	WED	W	10	41	
20210211	11.2.2021	11	th	5	Thursday	THU	T	11	42	
20210212	12.2.2021	12	th	6	Friday	FRI	F	12	43	
20210213	13.2.2021	13	th	7	Saturday	SAT	S	13	44	
20210214	14.2.2021	14	th	1	Sunday	SUN	S	14	45	
20210215	15.2.2021	15	th	2	Monday	MON	M	15	46	
20210216	16.2.2021	16	th	3	Tuesday	TUE	T	16	47	
20210217	17.2.2021	17	th	4	Wednesday	WED	W	17	48	
20210218	18.2.2021	18	th	5	Thursday	THU	T	18	49	
20210219	19.2.2021	19	th	6	Friday	FRI	F	19	50	
20210220	20.2.2021	20	th	7	Saturday	SAT	S	20	51	
20210221	21.2.2021	21	st	1	Sunday	SUN	S	21	52	
20210222	22.2.2021	22	nd	2	Monday	MON	M	22	53	
20210223	23.2.2021	23	rd	3	Tuesday	TUE	T	23	54	
20210224	24.2.2021	24	th	4	Wednesday	WED	W	24	55	
20210225	25.2.2021	25	th	5	Thursday	THU	T	25	56	
20210226	26.2.2021	26	th	6	Friday	FRI	F	26	57	
20210227	27.2.2021	27	th	7	Saturday	SAT	S	27	58	
20210228	28.2.2021	28	th	1	Sunday	SUN	S	28	59	
20210301	1.3.2021	1	st	2	Monday	MON	M	1	60	
20210302	2.3.2021	2	nd	3	Tuesday	TUE	T	2	61	
20210303	3.3.2021	3	rd	4	Wednesday	WED	W	3	62	

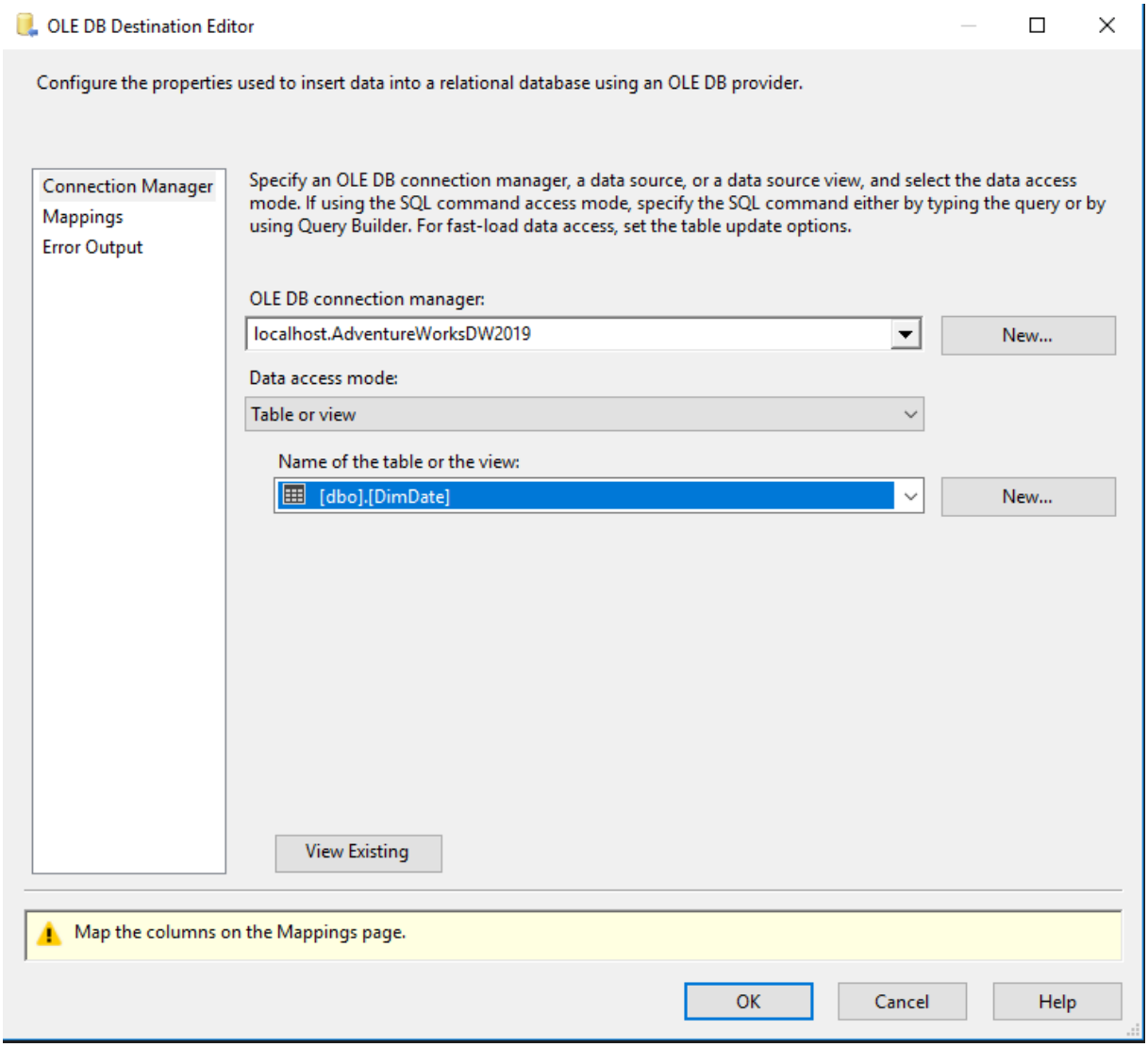
Slika 22. Vremenski podaci [28]

Nadalje, slika 23 prikazuje odabir Excel datoteke „DimDate“ kao izvor podataka.



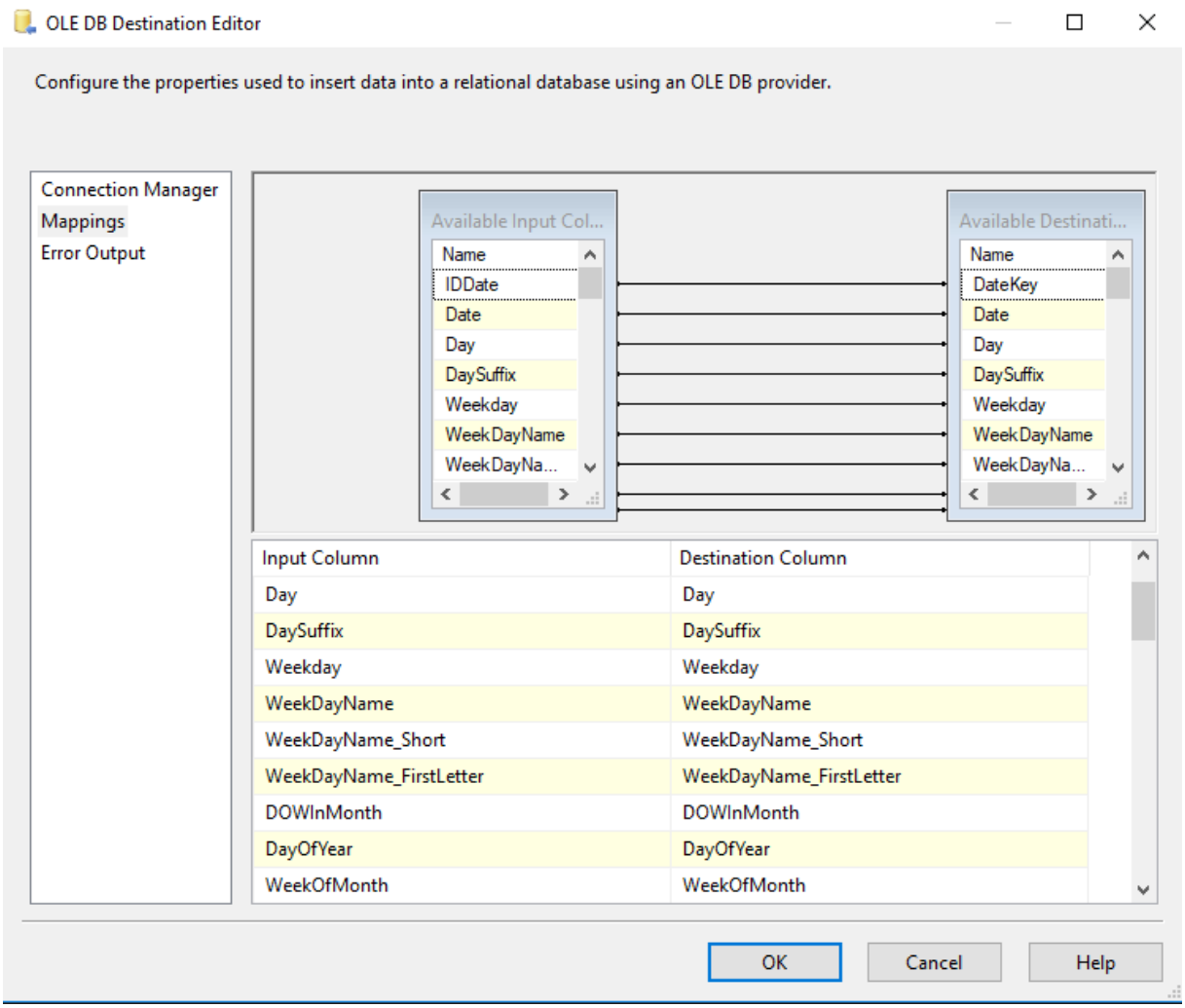
Slika 23. Odabir Excel datoteke sa vremenskim podacima [28]

Slika 24 prikazuje povezivanje sa skladištem podataka i željenom tablicom.



Slika 24. Odabir baze podataka i tablice [28]

Sljedeća slika, slika 25, prikazuje mapiranje stupaca iz Excel datoteke u vremensku dimenzijsku tablicu. Nakon mapiranja, sve je spremno za prijenos podataka u vremensku dimenzijsku tablicu.



Slika 25. Mapiranje stupaca između Excel datoteke i vremenske dimenzije [28]

4.5.4. Izrada činjenične tablice

Nakon kreiranja dimenzijskih tablica, sljedeći korak u ovom procesu je kreiranje činjenične tablice. Činjenična tablica je važna zbog toga što će se u njoj kao završni cilj naći nužni podaci za analizu kao što su strani ključevi na dimenzijske tablice („DimProduct“, „DimDate“, i „DimEmployee“) te cijena pojedinog proizvoda. Uz pomoć takvog prikaza moći će se jednostavno doći do odgovora na pitanje: Koliko je prodajni predstavnik xy zaradio novaca poduzeću prodajom dijelova za bicikle u određenom vremenskom razdoblju? Činjenična tablica „FactSales“ će se sastojati od stranih ključeva na dimenzijske tablice „DimEmployee“, „DimProduct“ i „DimDate“ te prodajnog iznosa. Slika 26 prikazuje naredbu za kreiranje

činjenične tablice uz pomoć koje će podatak, tj. vrijednosti u stupcu „Amount“, postati informacija, odnosno dobit će značenje u obliku odgovora na gore navedeno pitanje.

```
CREATE TABLE dbo.FactSales
(
    IDFactSale int IDENTITY(1, 1) NOT NULL
  , DateID int NOT NULL
  , ProductID int NOT NULL
  , EmployeeID int NOT NULL
  CONSTRAINT [PK_factSales_IDFactSale] PRIMARY KEY CLUSTERED (IDFactSale ASC)
) ON [PRIMARY];
```

Slika 26. Kreiranje tablice "FactSales" [28]

Slika 27 prikazuje kreiranje stranih ključeva na dimenzijske tablice „DimDate“, „DimEmployee“ i „DimProduct“.

```
ALTER TABLE [dbo].[FactSales] ADD CONSTRAINT [FK_FactSales_DimDate] FOREIGN KEY ([DateID])
REFERENCES [dbo].[DimDate] ([IDDate]);
GO

ALTER TABLE [dbo].[FactSales] ADD CONSTRAINT [FK_FactSales_DimEmployee] FOREIGN KEY ([EmployeeID])
REFERENCES [dbo].[DimEmployee] ([IDEmployee]);
GO

ALTER TABLE [dbo].[FactSales] ADD CONSTRAINT [FK_FactSales_DimProduct] FOREIGN KEY ([ProductID])
REFERENCES [dbo].[DimProduct] ([IDProduct]);
GO
```

Slika 27. Kreiranje stranih ključeva [28]

4.5.5. Punjenje činjenične tablice

Sljedeći korak je kreiranje međuskladišne činjenične tablice. U činjeničnoj tablici nalaze se strani ključevi na dimenzijske tablice. Ti ID-jevi su brojčani unutarnji podaci te u rijetkim slučajevima takvi ID-evi stignu s izvora podataka. Primjerice, proizvod naziva „AWC Logo Cap“ ima unutarnji ID jednak broju 3 te taj broj 3 moramo spremiti u činjeničnu tablicu. U većini slučajeva s izvora će doći podatak „AWC Logo Cap“, a ne broj 3 te je potrebno spremiti u međuskladišnu tablicu i informaciju o nazivu proizvoda koja se povezuje s podacima u dimenzijskim tablicama te se u činjeničnu tablicu sprema odgovarajući unutarnji ID. Odabir međuskladišnih stupaca ovisi o poznavanju izvora. Stupci u međuskladišnoj tablici „factSales“

će biti „ProductName“, „FirstName“, „LastName“ i „Amount“. Slika 28 prikazuje naredbu za kreiranje međuskladišne „factSales“ tablice.

```
CREATE TABLE staging.factSales
(
    IDStagingFactSale int IDENTITY (1,1) NOT NULL
    ProductName nvarchar(50) NOT NULL,
    EmployeeFirstName nvarchar(50) NOT NULL,
    EmployeeLastName nvarchar(50) NOT NULL,
    SellDate datetime NOT NULL,
    Amount float NOT NULL
 CONSTRAINT [PK_staging_factSales_IDStagingFactSale] PRIMARY KEY CLUSTERED (IDStagingFactSale ASC)
) ON [PRIMARY];
```

Slika 28. Kreiranje međuskladišne tablice "factSales" [28]

Nakon kreiranja međuskladišne činjenične tablice, slijedi punjenje iste kroz SSIS komponente. Izvori podataka će biti tekstualne, Excel i XML datoteke. Prvi korišteni izvor podataka za punjenje skladišta podataka će biti tekstualna datoteka. Međuskladišna činjenična tablica će se puniti kroz SSIS komponente koje uključuju komponentu za rad sa tekstualnim datotekama („Flat File Source“) i komponentu odredišta podataka („OLE DB Destination“) u kojoj se odabire tablica unutar skladišta podataka u koju želimo spremiti podatke iz tekstualne datoteke.

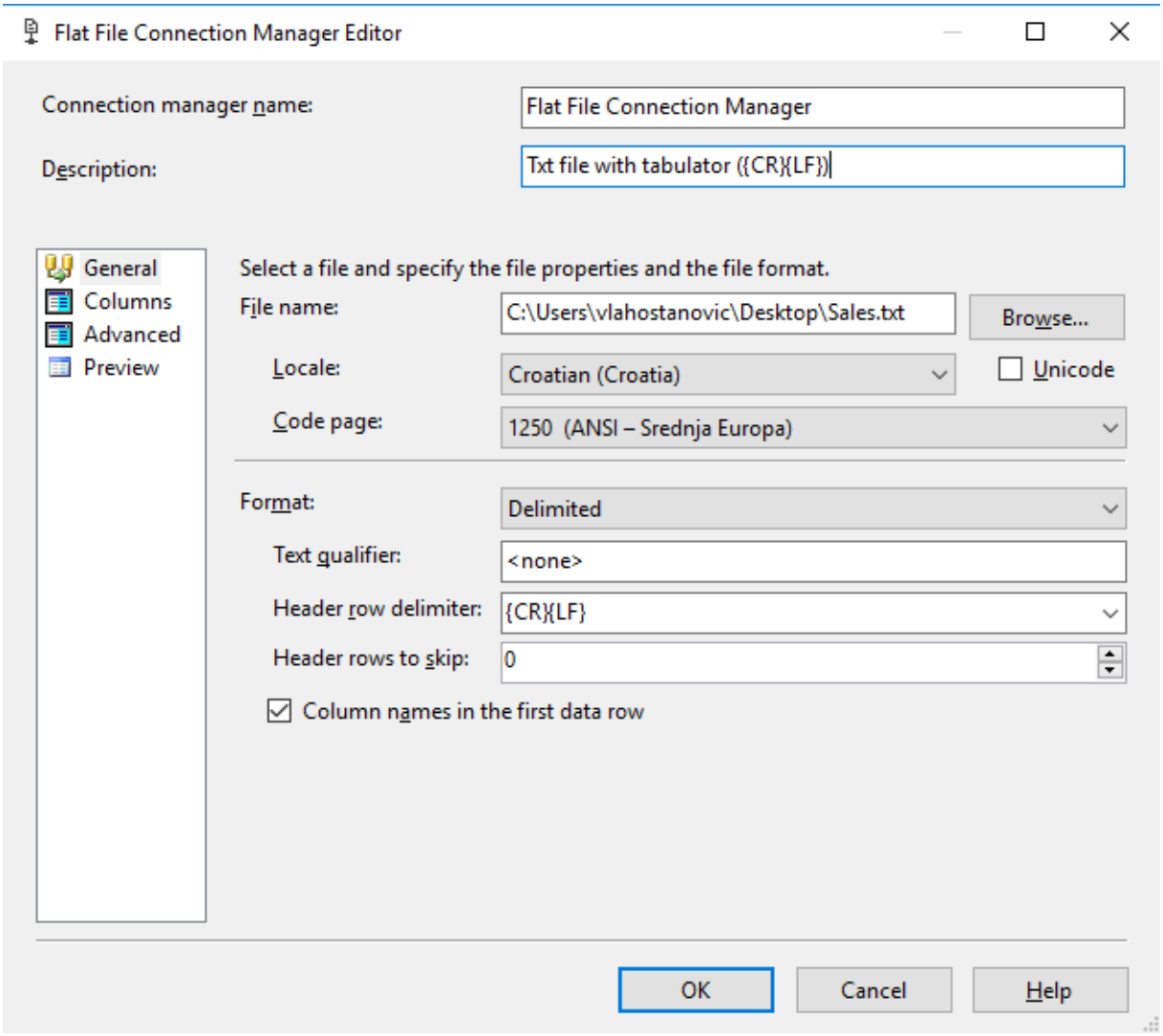
Tekstualna datoteka „Sales“ sadrži podatke o prodaji dijelova za bicikle. U stupcu „Name“ nalaze se podaci o nazivu dijelova za bicikle, stupci „First Name“ i „LastName“ sadrže ime i prezime zaposlenika, „SellDate“ stupac sadrži datum prodaje te stupac „Amount“ sadrži podatke o tome koliko je pojedini zaposlenik prodao određenih dijelova za bicikl u jednom danu. Nakon mapiranja stupaca iz tekstualne datoteke „Sales“ na stupce iz međuskladišne tablice „factSales“ u skladištu podataka AdventureWorksDW2019, podaci su spremni za prijenos u skladište podataka. Kako bi se moglo učitati podatke potrebno je mapirati stupce iz tekstualnog dokumenta na stupce međuskladišne tablice. Prethodno opisanim koracima podaci su se unijeli u međuskladišnu tablicu.

Slika 29 prikazuje podatke u tekstualnoj datoteci „Sales“.

```
Name    FirstName       LastName        SellDate        Amount
Bike Wash - Dissolver   Tsvi    Reiter  20130530        238,50
Bike Wash - Dissolver   Tete    Mensa-Annan     20130530        119,25
Bike Wash - Dissolver   Syed    Abbas   20130530        39,75
Bike Wash - Dissolver   Stephen Jiang   20130530        71,55
Bike Wash - Dissolver   Shu     Ito     20130530        214,65
Bike Wash - Dissolver   Ranjit  Varkey Chudukatil       20130530        254,40
Bike Wash - Dissolver   Rachel  Valdez  20130530        262,35
Bike Wash - Dissolver   Pamela  Ansman-Wolfe    20130530        71,55
Bike Wash - Dissolver   Michael Blythe  20130530        206,70
Bike Wash - Dissolver   Lynn    Tsoflias        20130530        159,00
Bike Wash - Dissolver   Linda   Mitchell        20130530        349,80
Bike Wash - Dissolver   José    Saraiva 20130530        262,35
Bike Wash - Dissolver   Jillian Carson  20130530        421,35
Bike Wash - Dissolver   Jae     Pak     20130530        341,85
Bike Wash - Dissolver   Garrett Vargas  20130530        214,65
Bike Wash - Dissolver   David   Campbell        20130530        79,50
Bike Wash - Dissolver   Amy     Alberts 20130530        23,85
Classic Vest, S Stephen Jiang   20130530        508,00
Classic Vest, S Shu     Ito     20130530        2222,50
```

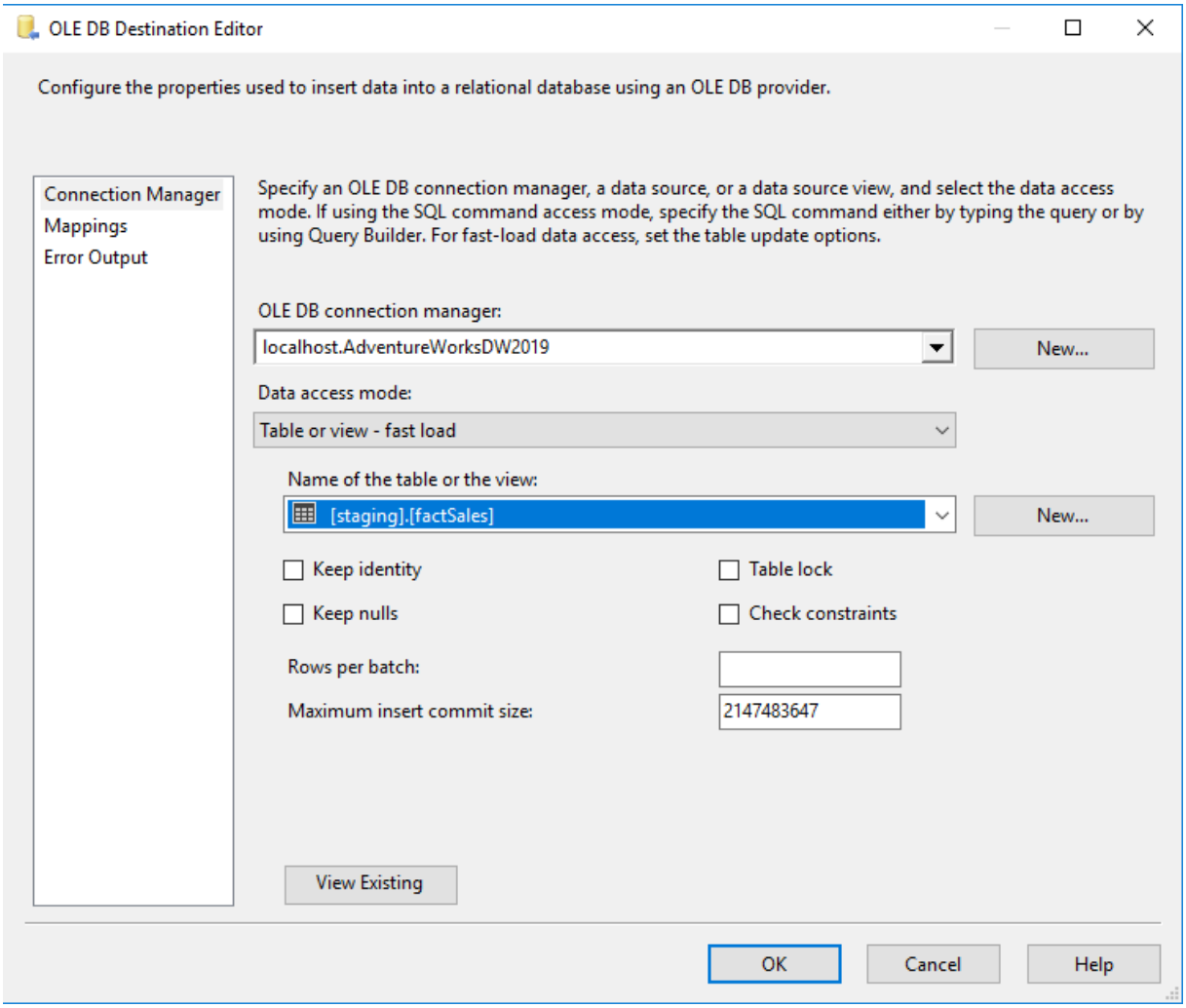
Slika 29. podaci u tekstualnoj datoteci [28]

Slika 30 prikazuje odabir tekstualne datoteke kao izvora podataka.



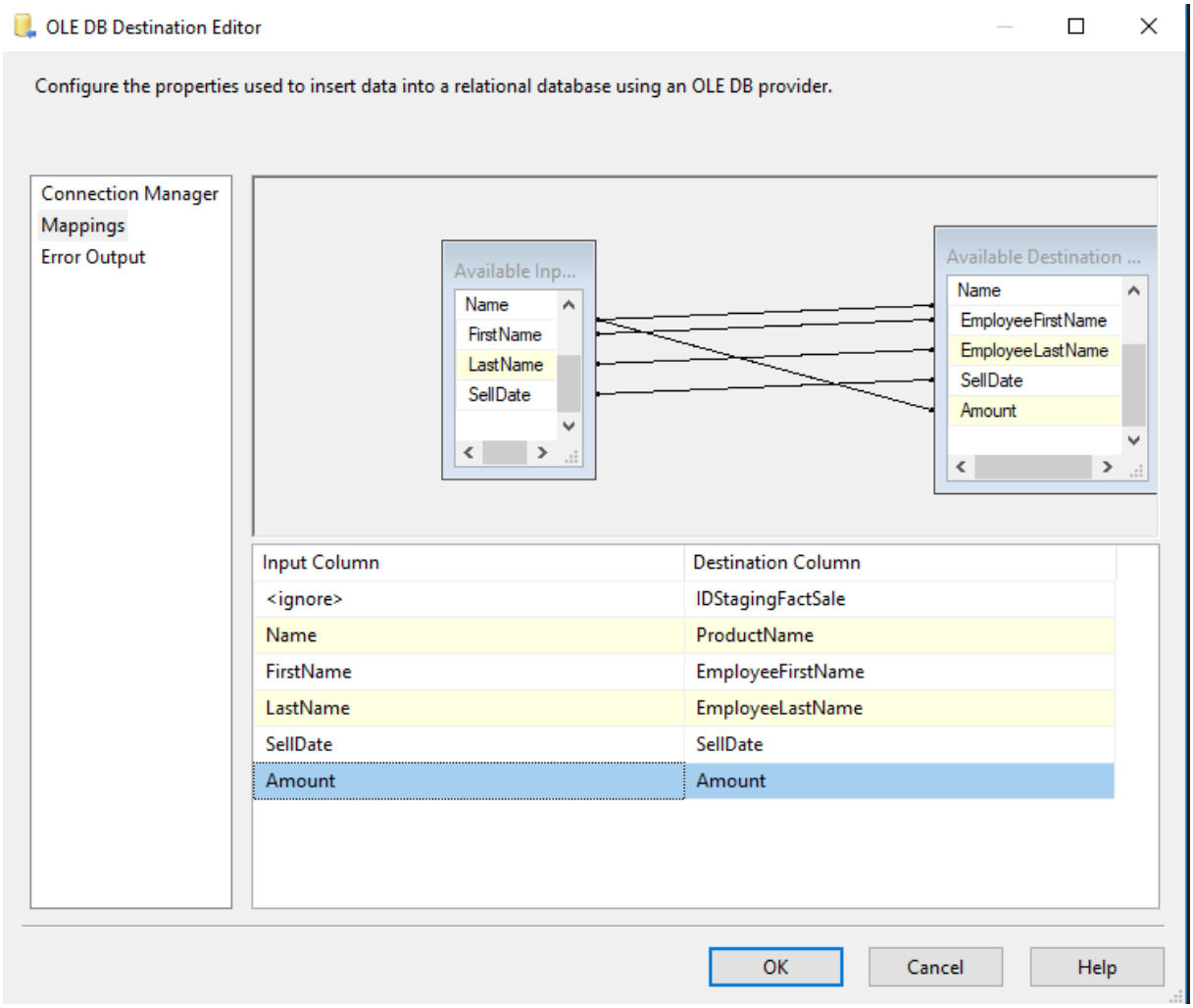
Slika 30. Odabir tekstualne datoteke kao izvora podataka [28]

Slika 31 prikazuje povezivanje sa skladištem podataka i željenom tablicom „factSales“.



Slika 31. Odabir baze podataka i tablice [28]

Sljedeća slika, slika 32, prikazuje mapiranje stupaca iz tekstualne datoteke „Sales“ u međuskladišnu činjeničnu tablicu. Nakon mapiranja, sve je spremno za prijenos podataka u skladište podataka.



Slika 32. Mapiranje stupaca između tekstualne datoteke i međuskladišne činjenične tablice [28]

Drugi izvor podataka za punjenje skladišta podataka će biti Excel datoteka. Međuskladišna činjenična tablica će se puniti kroz SSIS komponente koje uključuju komponentu za rad sa tekstualnim datotekama („Excel Source“) i komponentu odredišta podataka („OLE DB Destination“) u kojoj se odabire tablica unutar skladišta podataka u koju želimo spremiti podatke iz Excel datoteke. Excel datoteka „Sales“ sadrži podatke o prodaji dijelova za bicikle. U stupcu „Name“ nalaze se podaci o nazivu dijelova za bicikle, stupci „First Name“ i „LastName“ sadrže ime i prezime zaposlenika, „SellDate“ datum prodaje te stupac „Amount“ sadrži podatke o tome koliko je pojedini zaposlenik prodao određenih dijelova za bicikl u jednom danu. Nakon mapiranja stupaca iz Excel datoteke „Sales“ na stupce iz međuskladišne

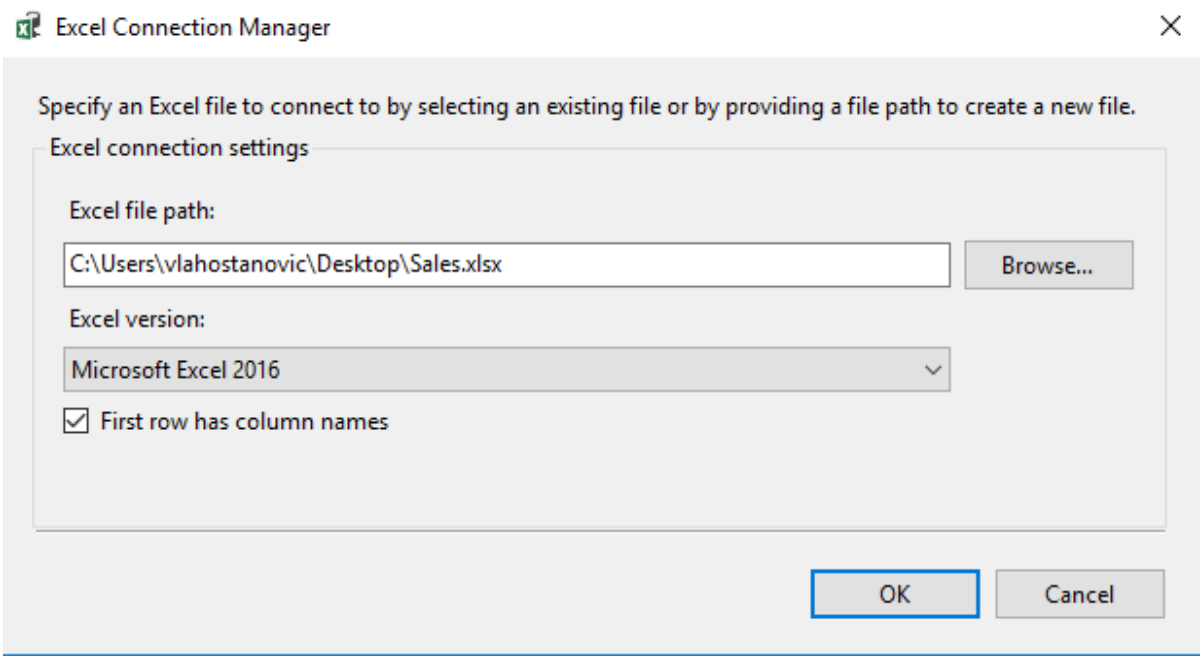
tablice „factSales“ u skladištu podataka AdventureWorksDW2019, podaci su spremni za prijenos u skladište podataka.

Slika 33 prikazuje izvorišne podatke u Excel datoteci „Sales“.

Chain	Linda	Mitchell	2013-05-30 00:00:00.00	404,8
Chain	José	Saraiva	2013-05-30 00:00:00.00	283,36
Chain	Jillian	Carson	2013-05-30 00:00:00.00	526,24
Chain	Jae	Pak	2013-05-30 00:00:00.00	546,48
Chain	Garrett	Vargas	2013-05-30 00:00:00.00	202,4
Chain	David	Campbell	2013-05-30 00:00:00.00	202,4
Chain	Amy	Alberts	2013-05-30 00:00:00.00	121,44
Classic Vest, S	Syed	Abbas	2013-05-30 00:00:00.00	444,5
Classic Vest, S	Tete	Mensa-Annan	2013-05-30 00:00:00.00	1206,5
Classic Vest, S	Tsvi	Reiter	2013-05-30 00:00:00.00	2095,5
Front Brakes	Amy	Alberts	2013-05-30 00:00:00.00	745,5
Front Brakes	David	Campbell	2013-05-30 00:00:00.00	1171,5
Front Brakes	Garrett	Vargas	2013-05-30 00:00:00.00	1278
Front Brakes	Jae	Pak	2013-05-30 00:00:00.00	2130
Front Brakes	Jillian	Carson	2013-05-30 00:00:00.00	2556
Front Brakes	José	Saraiva	2013-05-30 00:00:00.00	1597,5
Front Brakes	Linda	Mitchell	2013-05-30 00:00:00.00	3088,5
Front Brakes	Lynn	Tsoflias	2013-05-30 00:00:00.00	2875,5
Front Brakes	Michael	Blythe	2013-05-30 00:00:00.00	2769
Front Brakes	Pamela	Ansman-Wolfe	2013-05-30 00:00:00.00	1278
Front Brakes	Rachel	Valdez	2013-05-30 00:00:00.00	1278
Front Brakes	Ranjit	Varkey Chudukatil	2013-05-30 00:00:00.00	2875,5
Front Brakes	Shu	Ito	2013-05-30 00:00:00.00	1491
Front Brakes	Stephen	Jiang	2013-05-30 00:00:00.00	319,5
Front Brakes	Syed	Abbas	2013-05-30 00:00:00.00	426
Front Brakes	Tete	Mensa-Annan	2013-05-30 00:00:00.00	745,5
Front Brakes	Tsvi	Reiter	2013-05-30 00:00:00.00	1704
Front Derailleur	Amy	Alberts	2013-05-30 00:00:00.00	548,94
Front Derailleur	David	Campbell	2013-05-30 00:00:00.00	1097,88
Front Derailleur	Garrett	Vargas	2013-05-30 00:00:00.00	1006,39

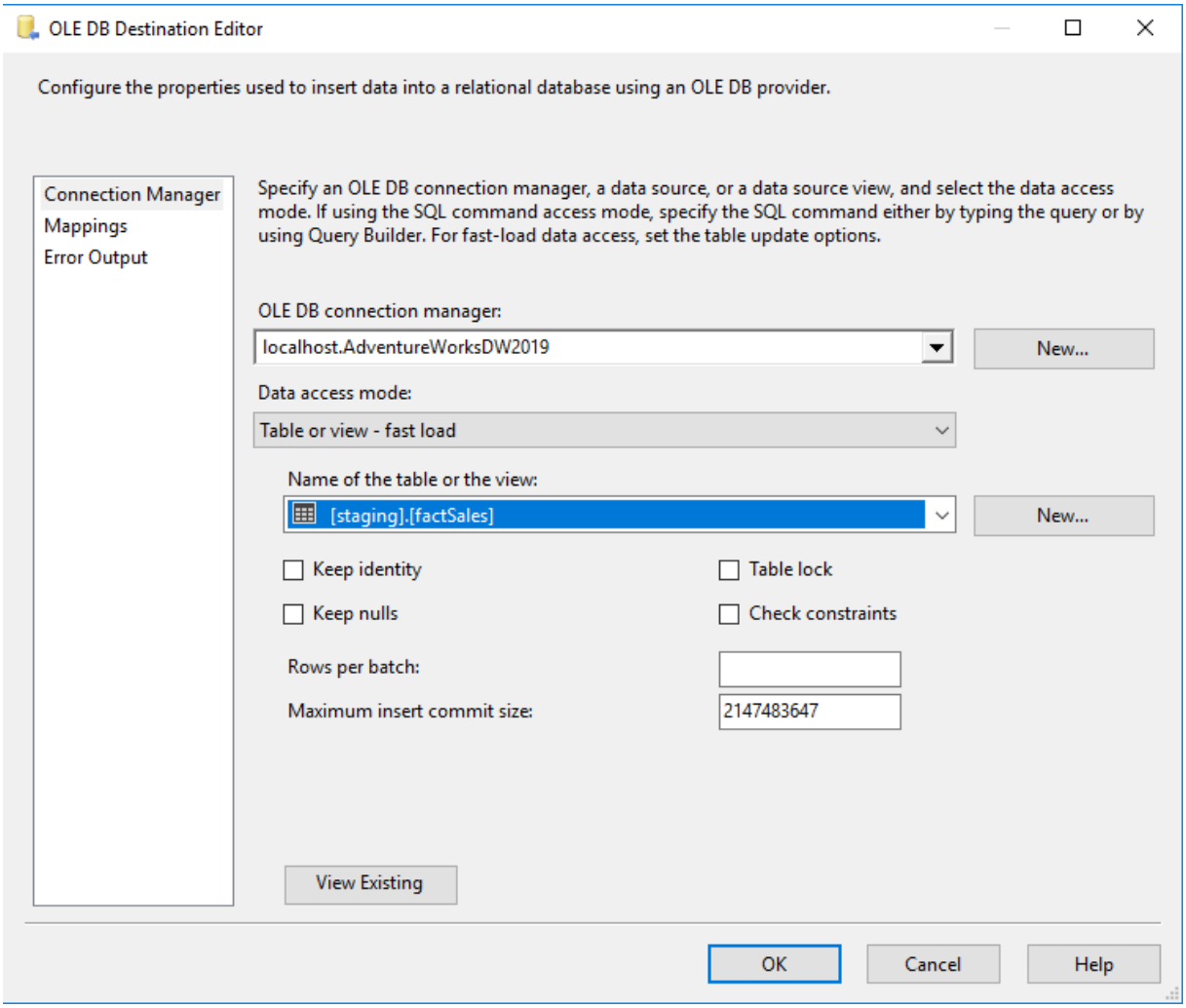
Slika 33. Podaci u Excel datoteci [28]

Slika 34 prikazuje odabir Excel datoteke kao izvor podataka.



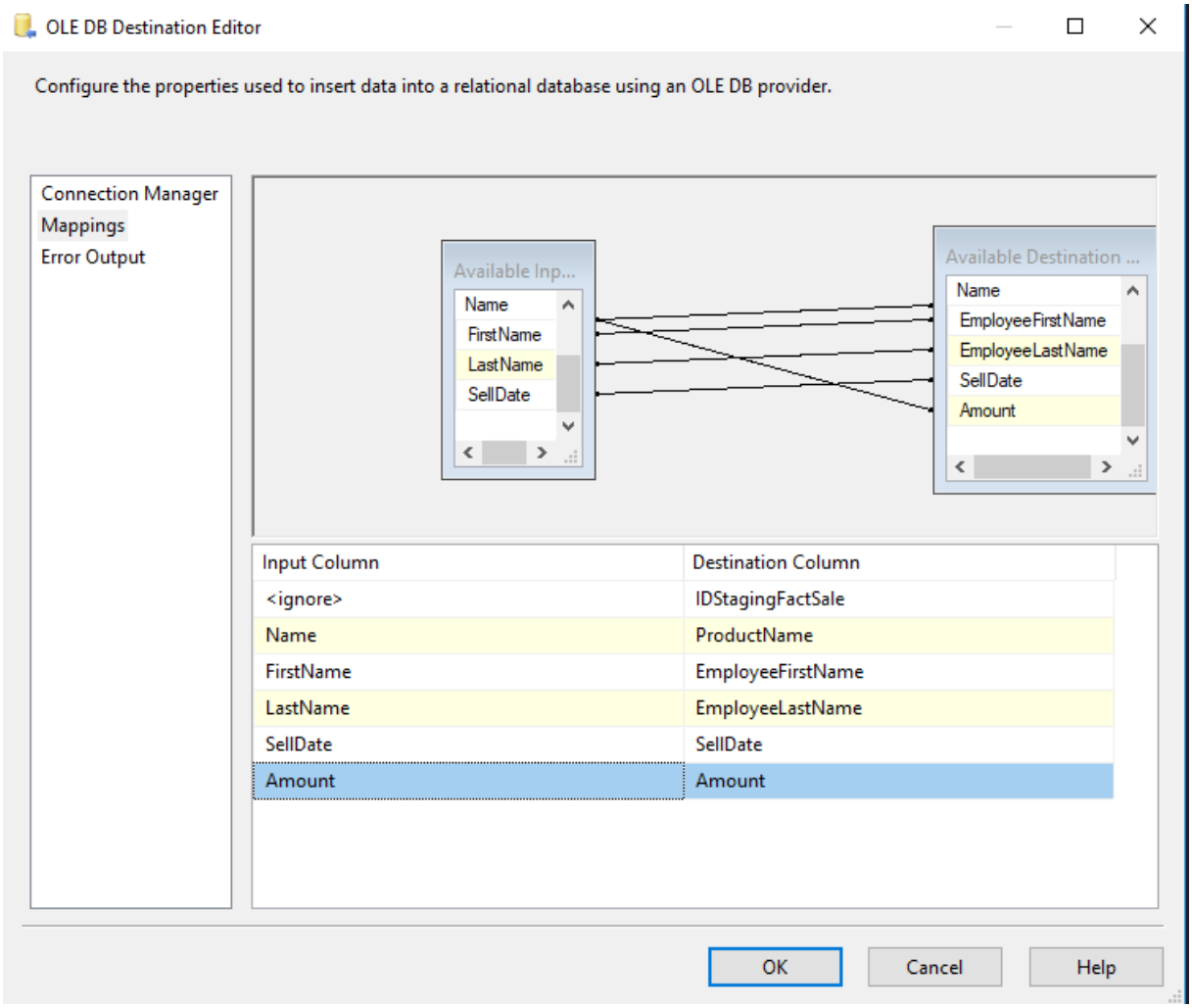
Slika 34. Excel datoteka kao izvor podataka [28]

Slika 35 prikazuje povezivanje sa skladištem podataka i željenom tablicom „factSales“.



Slika 35. Odabir baze podataka i tablice [28]

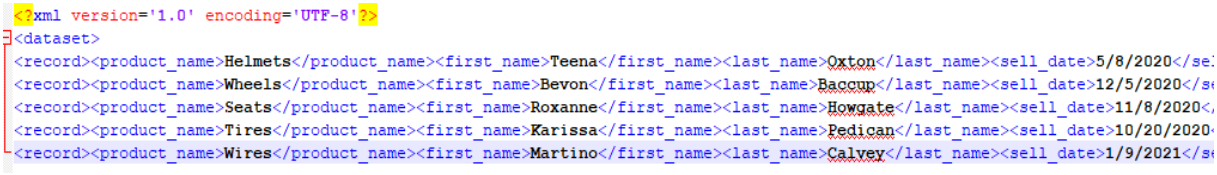
Sljedeća slika, slika 36, prikazuje mapiranje stupaca iz Excel datoteke u međuskladišnu činjeničnu tablicu. Nakon mapiranja, sve je spremno za prijenos podataka u skladište podataka.



Slika 36. Mapiranje stupaca između Excel datoteke i međuskladišne činjenične tablice [28]

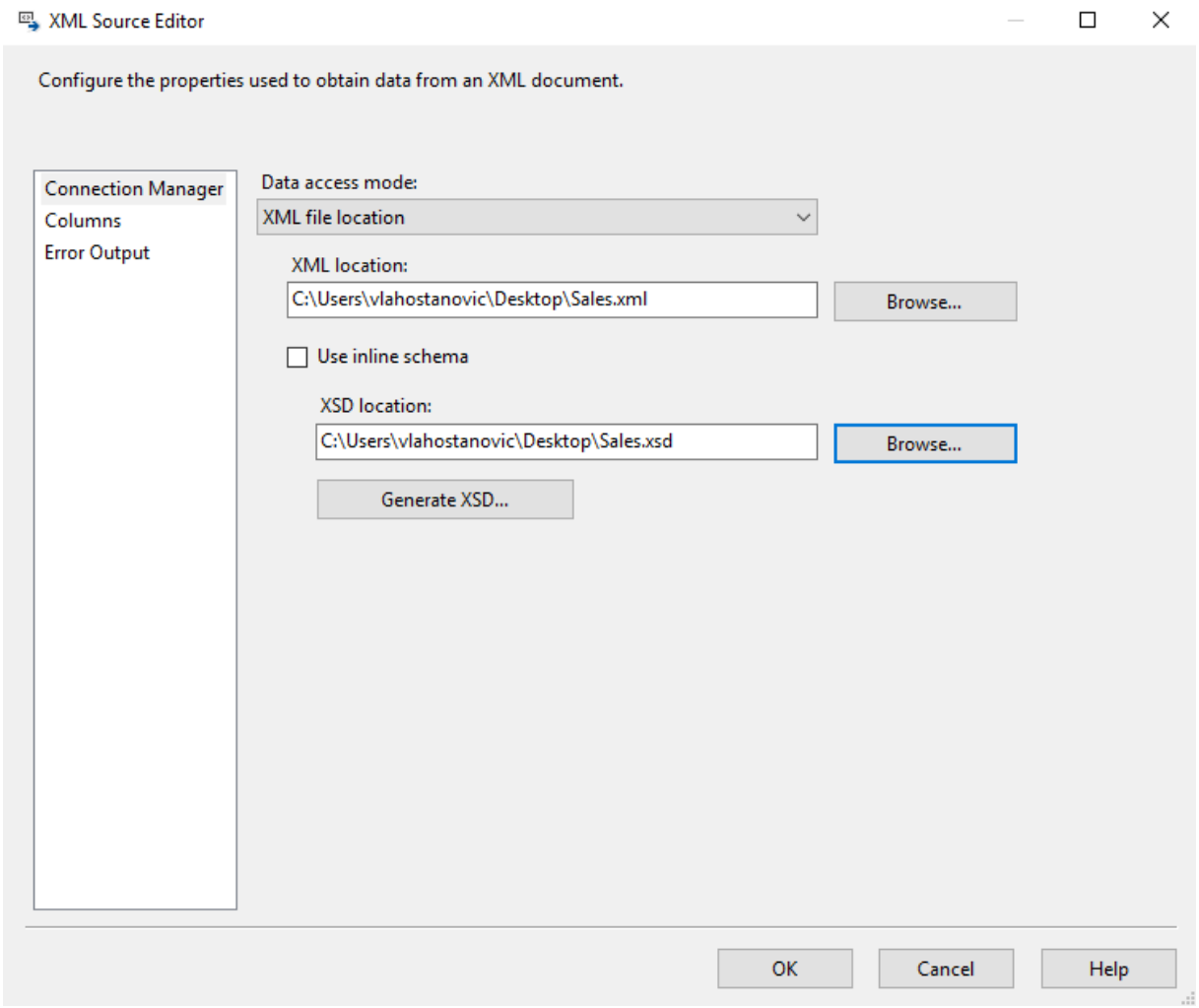
Treći korišteni izvor podataka je XML datoteka. Međuskladišna činjenična tablica će se puniti kroz SSIS komponente koje uključuju komponentu za rad sa XML datotekama („XML Source“) i komponentu odredišta podataka („OLE DB Destination“) u kojoj se odabire tablica unutar skladišta podataka u koju želimo spremiti podatke iz XML datoteke. XML datoteka „Sales“ sadrži podatke o prodaji dijelova za bicikle. U elementu „Name“ nalaze se podaci o nazivu dijelova za bicikle, elementi „First Name“ i „LastName“ sadrže ime i prezime zaposlenika, „SellDate“ datum prodaje te element „Amount“ sadrži podatke o tome koliko je pojedini zaposlenik prodao određenih dijelova za bicikl u jednom danu. Nakon mapiranja elemenata iz XML datoteke „Sales“ na stupce iz međuskladišne tablice „factSales“ u skladištu podataka AdventureWorksDW2019, podaci su spremni za prijenos u skladište podataka.

Slika 37 prikazuje podatke u XML datoteci.



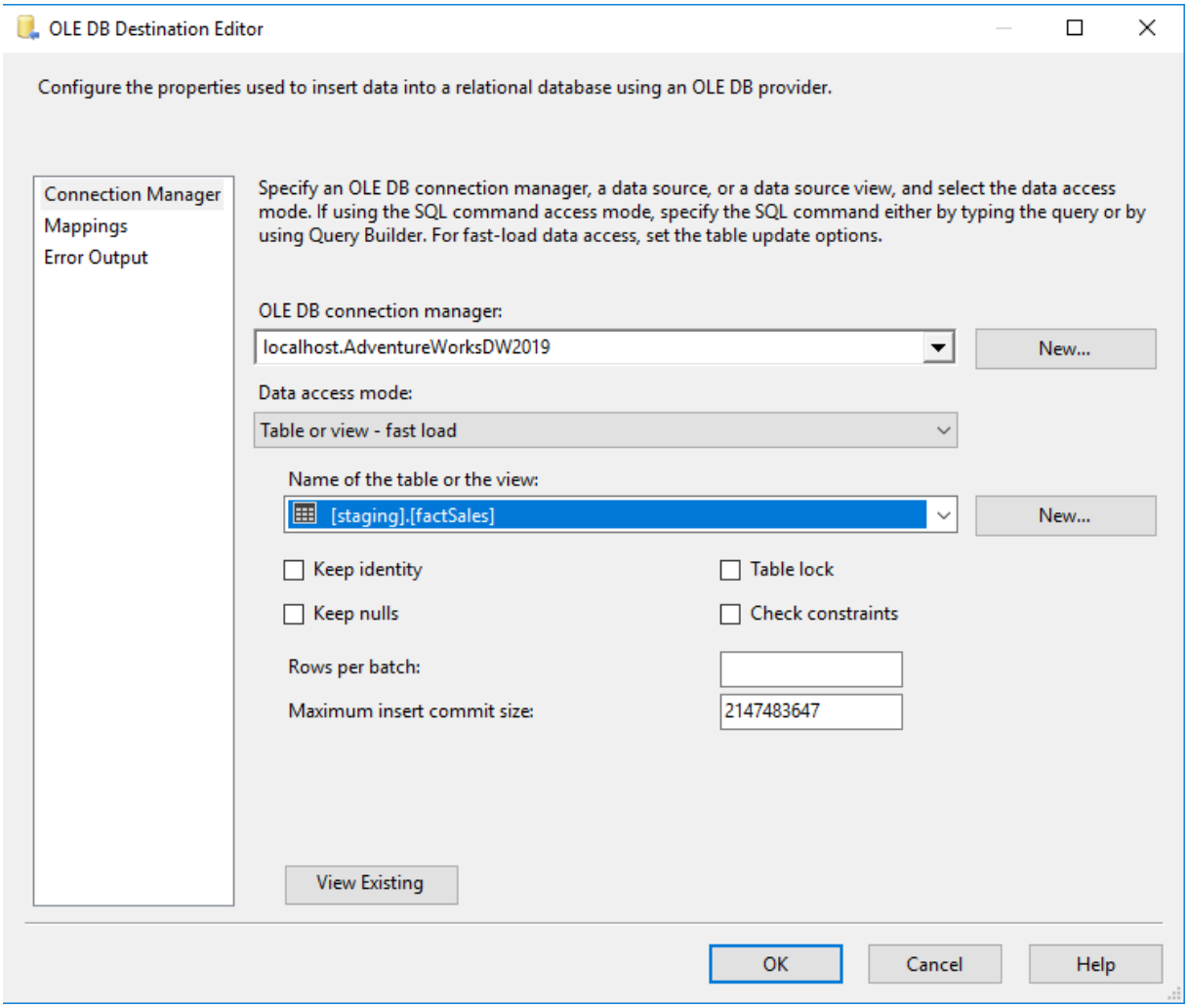
Slika 37. Podaci u XML datoteci [28]

Slika 38 prikazuje odabir XML datoteke kao izvora podataka.



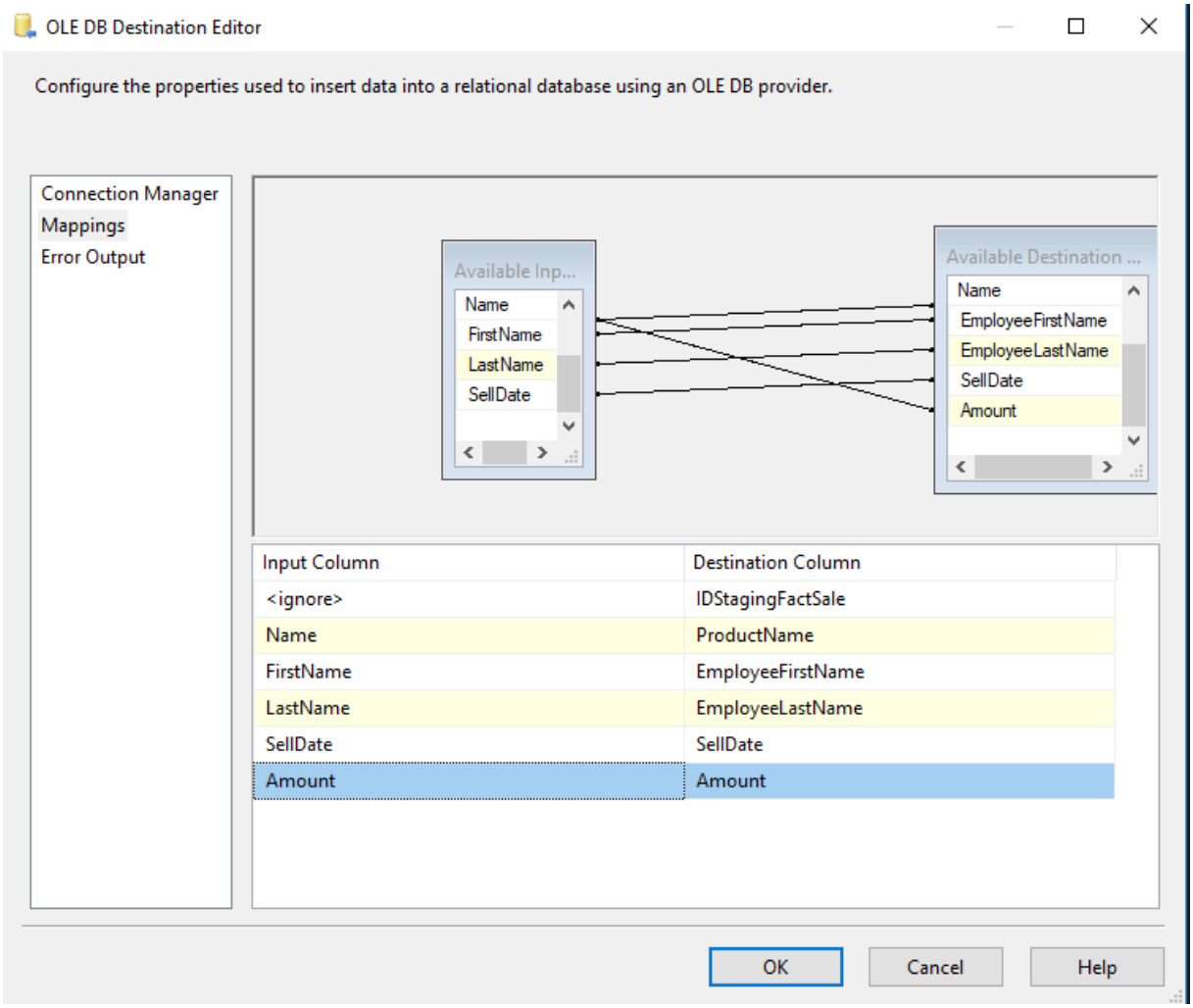
Slika 38. Prikaz izbora XML datoteke kao izvora podataka [28]

Slika 39 prikazuje povezivanje sa skladištem podataka i željenom tablicom.



Slika 39. Odabir baze podataka i tablice [28]

Slika 40 prikazuje mapiranje stupaca iz XML datoteke u međuskladišnu činjeničnu tablicu. Nakon mapiranja, sve je spremno za prijenos podataka u skladište podataka.



Slika 40. Mapiranje stupaca između XML datoteke i međuskladišne činjenične tablice [28]

Nakon što se međuskladišna činjenična tablica napunila podacima iz tri različita izvora podataka, potrebno je podatke dodati u činjeničnu tablicu. Proces će se obaviti procedurom tj. merge operacijom. Potrebno je obaviti tri glavne operacije u proceduri kako bi skladište podataka zadržalo funkcionalnost. Prva operacija za cilj ima pronaći odgovarajuće ID-jeve za njihovo spremanje u činjeničnu tablicu putem veze između stupaca međuskladišne činjenične tablice i dimenzijskih tablica. Druga operacija se odnosi na sumiranje prodajne cijene proizvoda tj. grupiranje po nazivu proizvoda, imenu i prezimenu zaposlenika te datumu kako bi za cilj jedan redak u činjeničnoj tablici označavao prodaju pojedinog proizvoda po zaposleniku na jedan dan u godini. Treća i posljednja korištena operacija odnosi se na pretvorbu datuma prodaje u datum koji ima isti format kao i primarni ključ u tablici „DimDate“.

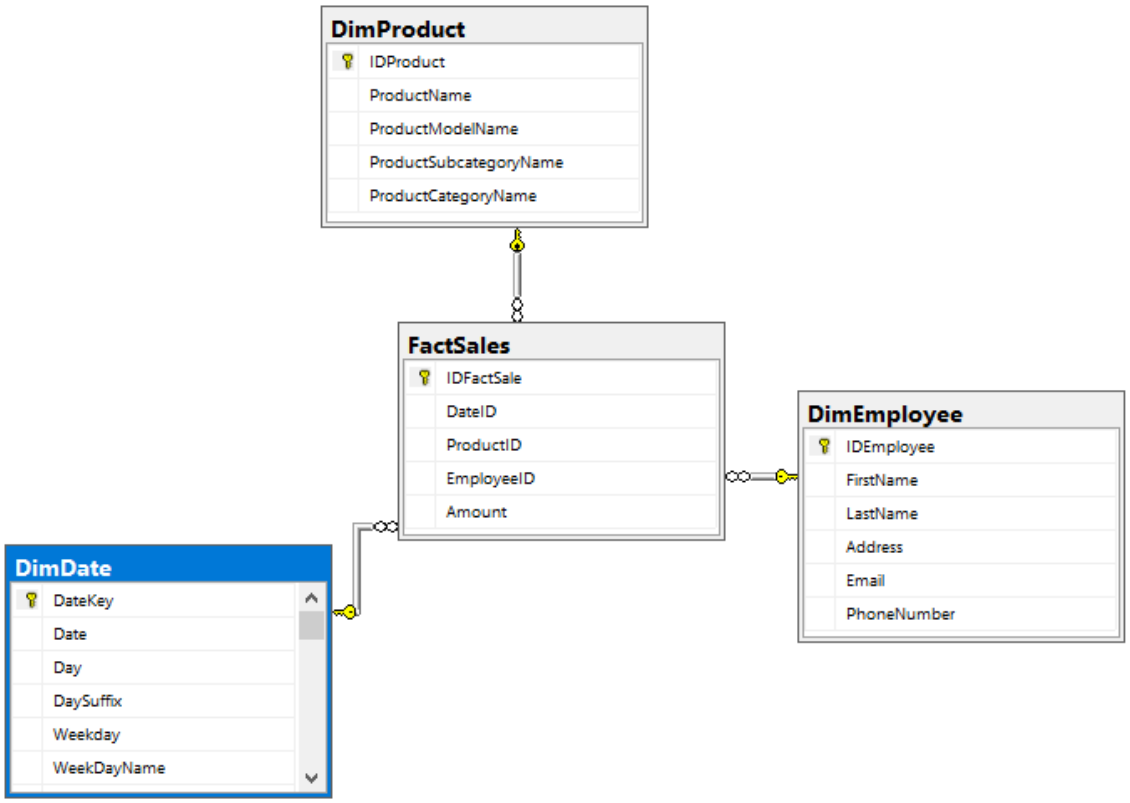
Slika 41 prikazuje naredbu za prijenos podataka iz međuskladišne činjenične tablice u činjeničnu tablicu.

```
CREATE PROCEDURE dbo.MergeFactSales
AS
BEGIN
MERGE dbo.factSales AS trg
USING ( SELECT dp.ProductID, de.IDEmployee, dd.IDDate, SUM(Amount) AS Amount FROM staging.factSales sf
        INNER JOIN dbo.DimEmployee AS de ON de.FirstName = sf.EmployeeFirstName AND de.LastName = sf.EmployeeLastName
        INNER JOIN dbo.DimProduct AS dp ON dp.ProductName = sf.ProductName
        INNER JOIN dbo.DimDate AS dd ON dd.IDDate = CONVERT( nvarchar(10), sf.SellDate, 112)
        GROUP BY dp.ProductID, de.IDEmployee, dd.IDDate
      ) src ON src.ProductID = trg.ProductID AND src.IDEmployee = trg.EmployeeID AND src.DateID = trg.DateID
WHEN NOT MATCHED BY TARGET
INSERT INTO dbo.factSales (DateID, ProductID, EmployeeID)
SELECT src.DateID, src.ProductID, src.EmployeeID;
END
```

Slika 41. Naredba za prijenos podataka [28]

4.5.6. Prikaz upita u skladištu podataka

Nakon prethodno opisanog koraka, skladište sadrži sve podatke koji su nužni kako bi se odgovorilo na pitanje kolika je prodaja dijelova za bicikle po prodajnom predstavniku u određenom vremenskom okviru. Ovim pitanjem se započeo ovaj praktični dio rada, a tablica niže predstavlja skup podataka pomoću kojih se ovo postavljeno pitanje može odgovoriti. Slika 42 prikazuje dijagram gotovog zvjezdastog modela skladišta podataka.



Slika 42. Prikaz zvjezdastog modela skladišta podataka [28]

Slika 43. prikazuje učitane podatke u tablici „DimProduct“.

ProductID	ProductName	ProductNumber	ProductModelName	ProductionSubcategoryName	ProductCategoryName
680	HL Road Frame - Black, 58	FR-R92B-58	HL Road Frame	Road Frames	Components
706	HL Road Frame - Red, 58	FR-R92R-58	HL Road Frame	Road Frames	Components
707	Sport-100 Helmet, Red	HL-U509-R	Sport-100	Helmets	Accessories
708	Sport-100 Helmet, Black	HL-U509	Sport-100	Helmets	Accessories
709	Mountain Bike Socks, M	SO-B909-M	Mountain Bike Socks	Socks	Clothing
710	Mountain Bike Socks, L	SO-B909-L	Mountain Bike Socks	Socks	Clothing
711	Sport-100 Helmet, Blue	HL-U509-B	Sport-100	Helmets	Accessories
712	AWC Logo Cap	CA-1098	Cycling Cap	Caps	Clothing
713	Long-Sleeve Logo Jersey, S	LJ-0192-S	Long-Sleeve Logo Jersey	Jerseys	Clothing
714	Long-Sleeve Logo Jersey, M	LJ-0192-M	Long-Sleeve Logo Jersey	Jerseys	Clothing
715	Long-Sleeve Logo Jersey, L	LJ-0192-L	Long-Sleeve Logo Jersey	Jerseys	Clothing
716	Long-Sleeve Logo Jersey, XL	LJ-0192-X	Long-Sleeve Logo Jersey	Jerseys	Clothing
717	HL Road Frame - Red, 62	FR-R92R-62	HL Road Frame	Road Frames	Components
718	HL Road Frame - Red, 44	FR-R92R-44	HL Road Frame	Road Frames	Components

Slika 43. tablica "DimProduct" [28]

Slika 44 prikazuje učitane podatke u tablici „DimEmployee“.

FirstName	LastName	AddressLine1	EmailAddress	PhoneNumber
Syed	Abbas	7484 Roundtree Drive	syed0@adventure-works.com	926-555-0182
David	Campbell	2284 Azalea Avenue	david8@adventure-works.com	740-555-0182
Garrett	Vargas	10203 Acorn Avenue	garrett1@adventure-works.com	922-555-0165
Tsvi	Reiter	8291 Crossbow Way	tsvi0@adventure-works.com	664-555-0112
Jillian	Carson	80 Sunview Terrace	jillian0@adventure-works.com	517-555-0117
Michael	Blythe	8154 Via Mexico	michael9@adventure-works.com	257-555-0154
Rachel	Valdez	Pascalstr 951	rachel0@adventure-works.com	1 (11) 500 555-0140
Amy	Alberts	5009 Orange Street	amy0@adventure-works.com	775-555-0164
Linda	Mitchell	2487 Riverside Drive	linda3@adventure-works.com	883-555-0116
José	Saraiva	9100 Sheppard Avenue North	josé1@adventure-works.com	185-555-0169
Jae	Pak	Downshire Way	jae0@adventure-works.com	1 (11) 500 555-0145
Ranjit	Varkey Chudukatil	94, rue Descartes	ranjit0@adventure-works.com	1 (11) 500 555-0117
Stephen	Jiang	2427 Notre Dame Ave.	stephen0@adventure-works.com	238-555-0197
Lynn	Tsoflias	34 Waterloo Road	lynn0@adventure-works.com	1 (11) 500 555-0190

Slika 44. tablica "DimEmployee" [28]

Slika 45 prikazuje rezultat postavljenog upita na početku rada.

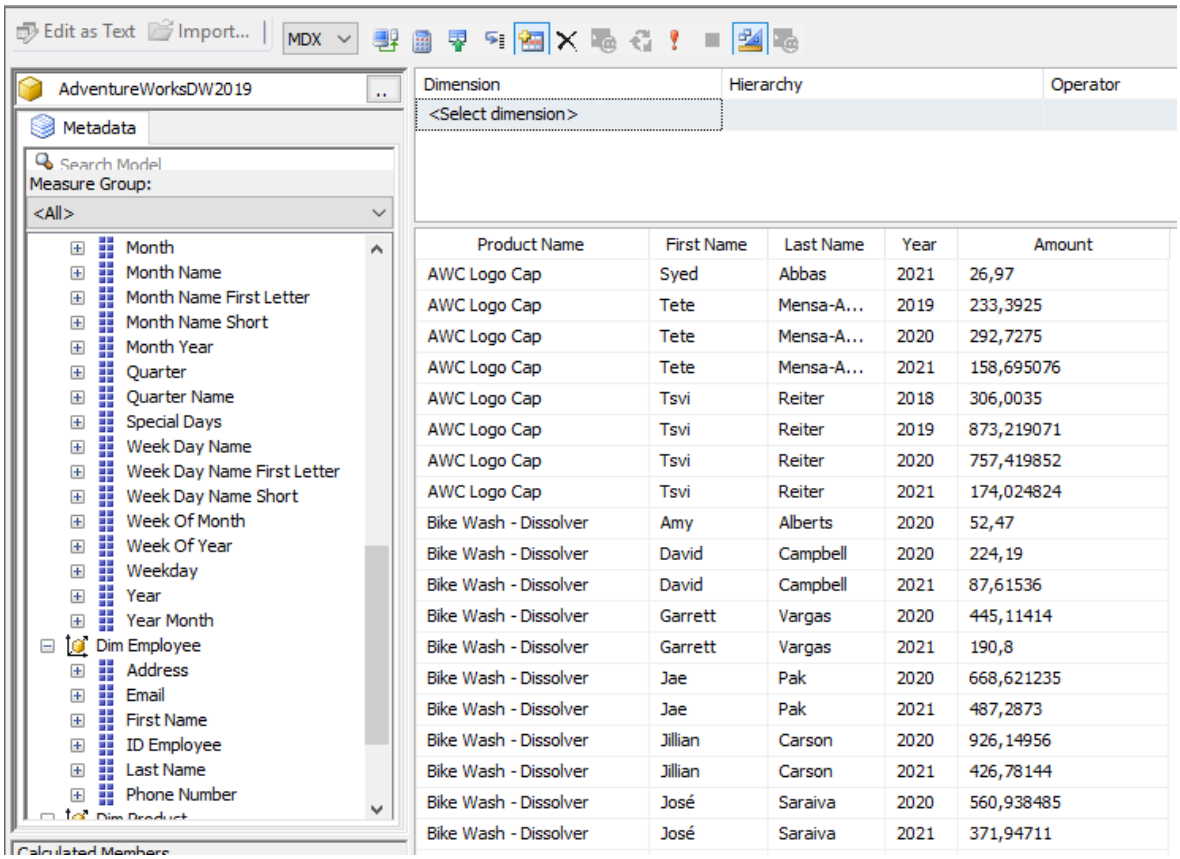
Name	FirstName	LastName	SellDate	Amount
Bike Wash - Dissolver	Jillian	Carson	20130530	421,35
Bike Wash - Dissolver	Jae	Pak	20130530	341,85
Bike Wash - Dissolver	Garrett	Vargas	20130530	214,65
Bike Wash - Dissolver	David	Campbell	20130530	79,50
Bike Wash - Dissolver	Amy	Alberts	20130530	23,85
Classic Vest, S	Stephen	Jiang	20130530	508,00
Classic Vest, S	Shu	Ito	20130530	2222,50
Classic Vest, S	Ranjit	Varkey Chudukatil	20130530	2476,50
Classic Vest, S	Rachel	Valdez	20130530	2349,50
Classic Vest, S	Pamela	Ansman-Wolfe	20130530	381,00
Classic Vest, S	Michael	Blythe	20130530	2476,50
Classic Vest, S	Lynn	Tsoflias	20130530	1905,00

Slika 45. rezultat upita o rezultatima prodaje [28]

5. PRIKAZ REZULTATA U VIŠEDIMENZIONALNOM MODELU

Skladište podataka je u svojoj osnovi baza podataka za izvještavanje. Da bi se mogao prikazati cjelokupni proces izrade skladišta podataka potrebno je prikazati i mogućnosti izvještavanja. U prethodno opisanom poglavlju prikazan je upit kojim se došlo do potrebnih podataka za analizu. Osim prikaza podataka, potrebno je dodatno pružiti mogućnost krajnjim korisnicima, odnosno menadžerima i voditeljima odjela ili poduzeća, da imaju pristup ovim podacima. U ovom dijelu rada prikazat će se izrada kocke koja je bitna za izvještavanje te izvještaji unutar Microsoftovog Power BI-a.

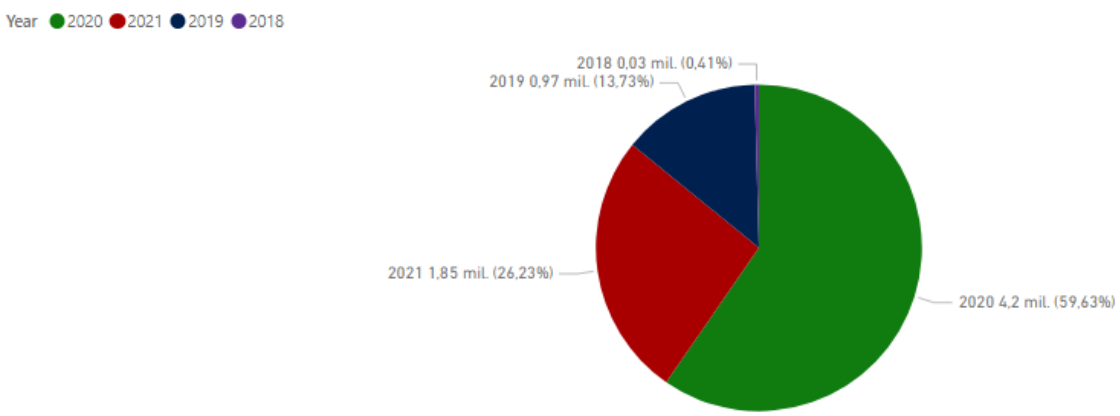
Kocka će se izgraditi od sljedećih tablica: "DimDate", "DimProduct", "DimEmployee" i "FactSales". Kreiranje kocke se obavlja unutar Visual Studija. S obzirom da su veze između tablica točno definirane, kreiranje kocke će SSAS automatski obaviti. Nakon kreiranja i naseljavanja kocke na Analysis server, u SSMS-u moguće je pristupiti kreiranoj kocki. Slika 46 prikazuje rezultat upita „Kolika je prodaja dijelova za bicikle po prodajnom predstavniku u određenom vremenskom okviru?" na Analysis serveru.



Product Name	First Name	Last Name	Year	Amount
AWC Logo Cap	Syed	Abbas	2021	26,97
AWC Logo Cap	Tete	Mensa-A...	2019	233,3925
AWC Logo Cap	Tete	Mensa-A...	2020	292,7275
AWC Logo Cap	Tete	Mensa-A...	2021	158,695076
AWC Logo Cap	Tsvi	Reiter	2018	306,0035
AWC Logo Cap	Tsvi	Reiter	2019	873,219071
AWC Logo Cap	Tsvi	Reiter	2020	757,419852
AWC Logo Cap	Tsvi	Reiter	2021	174,024824
Bike Wash - Dissolver	Amy	Alberts	2020	52,47
Bike Wash - Dissolver	David	Campbell	2020	224,19
Bike Wash - Dissolver	David	Campbell	2021	87,61536
Bike Wash - Dissolver	Garrett	Vargas	2020	445,11414
Bike Wash - Dissolver	Garrett	Vargas	2021	190,8
Bike Wash - Dissolver	Jae	Pak	2020	668,621235
Bike Wash - Dissolver	Jae	Pak	2021	487,2873
Bike Wash - Dissolver	Jillian	Carson	2020	926,14956
Bike Wash - Dissolver	Jillian	Carson	2021	426,78144
Bike Wash - Dissolver	José	Saraiva	2020	560,938485
Bike Wash - Dissolver	José	Saraiva	2021	371,94711

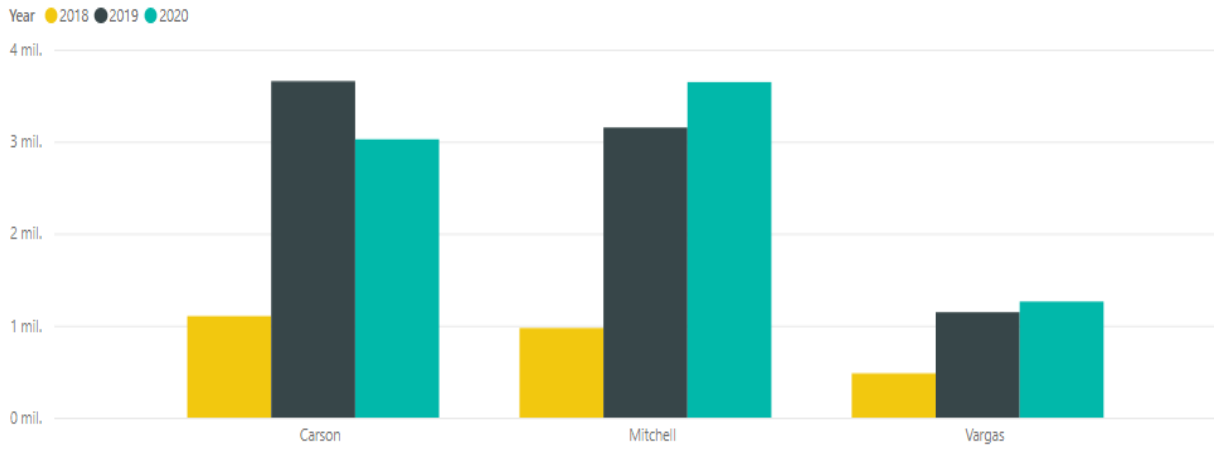
Slika 46. Prikaz višedimenzionalnog modela na *Analysis* serveru [28]

Osim gore navedenog prikaza podataka, prilikom izvještavanja moguće je uz pomoć raznih alata, poput Microsoft Power BI-a, kreirati grafičke prikaze pojedinih podataka koji bi bili važni krajnjim korisnicima za razne analize. Slika 47 prikazuje ukupnu prodaju poduzeća Adventure Works Cycles po godinama.



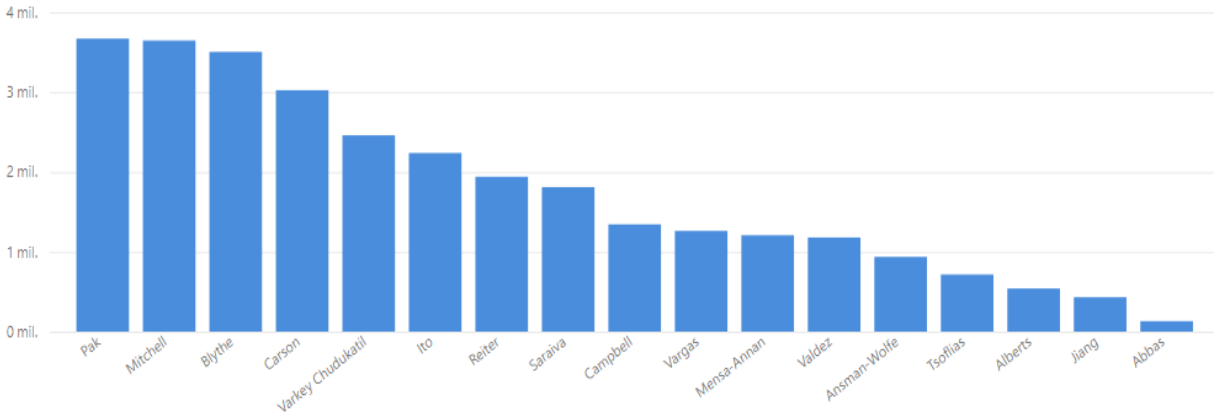
Slika 47. Ukupna prodaja po godinama [28]

Nadalje, slika 48 prikazuje prodaju koju su ostvarila tri zaposlenika po godinama. Za ovaj primjer, tri zaposlenika su predstavljena, a ako bi krajnji korisnici imali potrebu za pregledom svih zaposlenika i njihovog prodajnog učinka, i taj podatak se može lako prikazati u ovom programu za izvještavanje.



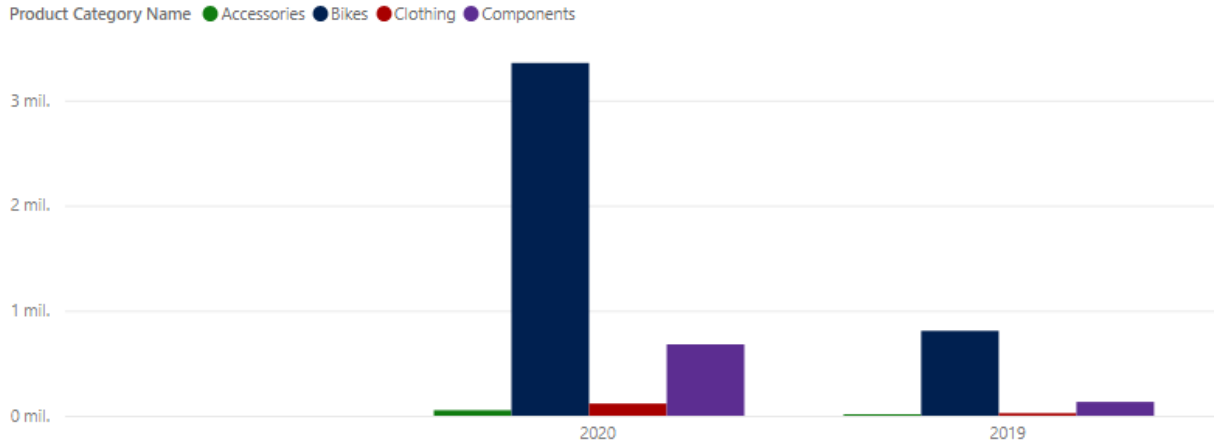
Slika 48. Prodaja zaposlenika po godinama [28]

Slika 49 za fokus uzima godinu 2020 i ilustrira koliko je pojedini zaposlenik firme ostvario firmi prihoda prodajom bicikala i biciklističke opreme.



Slika 49. Prodaja po zaposleniku 2020. godine. [28]

Nadalje, slika 50 prikazuje prodaju po kategoriji proizvoda za dvije godine (godinu 2019. i 2020.).



Slika 50. Prodaja po kategoriji proizvoda [28]

6. ZAKLJUČAK

Svjedoci smo da je u današnje vrijeme potreba za skladištenjem podataka veća nego ikad prije. Sva poduzeća današnjice, bez obzira na njihovu veličinu, dnevno generiraju veliku količinu podataka. Ti podaci su im neophodni kako za svakodnevne aktivnosti, tako i za dugoročno planiranje i analizu poslovnih odluka. Cilj ovog rada je bio da se tema skladištenja podataka obradi u cijelosti te da se na praktičnom primjeru pokaže izgradnja jednostavnog skladišta, počevši od podataka u relacijskoj bazi podataka i heterogenim izvorima do krajnjeg skupa podataka za analizu koju vrše klijenti odnosno poslovni korisnici. Detaljnim prikazom procesa skladištenja podataka kroz jednostavan primjer, čitatelj može shvatiti što stoji iza ovog procesa i koje korake je nužno slijediti kako bi se izgradilo, napunilo podacima i analiziralo jedno skladište podataka.

U radu su korištene Microsoftove suvremene tehnologije i alati za kreiranje jednostavnog skladišta podataka. Ponuđene Microsoftove opcije za rad s podacima su dovoljno razvijene i napredne te nailazak na neki manji izazov ne iziskuje od programera ulaganje velikog broja sati u rješavanje istog. Nadalje, Microsoftovi alati su raširenі i često korišteni u IT svijetu te su pomoć i rješenja za učestale izazove s kojima se programer može susresti prilikom svojeg rada sa skladištem podataka, dostupni i razrađeni *online*.

Praktični dio ovog rada prikazuje rad sa idealno strukturiranim datotekama i podacima. Bez obzira na jednostavnost podataka, praktični dio rada je ispunio svoj cilj koji se odnosio na prikaz putanje jednog podatka iz Excel, XML i tekstualne datoteke do konačnog odredišta tj. grafičkog prikaza analize podataka pogodnog poslovnim korisnicima.

U procesu izgradnje skladištenja podataka najzahtjevniji dio je zasigurno modeliranje skladišta podataka, odnosno odabir potrebnih podataka dostupnih iz raznih izvora. U ovom procesu veoma je bitno da programer dobro i kvalitetno shvati želje klijenta te da osigura pokrivenost svih zahtjeva od strane klijenta. U samoj izgradnji skladišta podataka, veliku ulogu igra i iskustvo programera, kako u njegovom radu, tako i u komunikaciji s klijentom.

LITERATURA

[1] K. Ćurko, "SKLADIŠTE PODATAKA- SUSTAV ZA POTPORU ODLUČIVANJU", Ekonomski pregled, vol.52, br. 7-8, str. 840-855, 2001. [Online]. Dostupno na: https://hrcak.srce.hr/28761. (pristupljeno 25.03.2021).

[2] W. H. Inmon, Building the data warehouse, 4th ed. Indianapolis, Ind: Wiley, 2005.

[3] K. D. Foote, "A Brief History of the Data Warehouse," DATAVERSITY, Apr. 19, 2018. Dostupno na: https://www.dataversity.net/brief-history-data-warehouse/ (pristupljeno 27.03.2021).

[4] W. H. Inmon, D. Strauss, and G. Neushloss, DW 2.0: the architecture for the next generation of data warehousing. Amsterdam ; Boston: Morgan Kaufmann, 2008.

[5] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 7th edition. McGraw-Hill Education, 2020.

[6] "What is the database architecture in DBMS? - Quora." Dostupno na: https://www.quora.com/What-is-the-database-architecture-in-DBMS (pristupljeno 29.4.2021).

[7] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, and B. Becker, The Data Warehouse Lifecycle Toolkit. Hoboken: John Wiley & Sons, 2011.

[8] M. Breslin, "DW MODELS Data Warehousing Battle of the Giants : Comparing the Basics of the Kimball and Inmon Models,"2004. Dostupno na: https://www.semanticscholar.org/paper/DW-MODELS-Data-Warehousing-Battle-of-the-Giants-%3A-Breslin/c80f8aaea5bf58846b0125b460401fed8230c2d2 (pristupljeno 26.03. 2021).

[9] "Data Mart vs. Data Warehouse," Panoply. Dostupno na: https://panoply.io/data-warehouse-guide/data-mart-vs-data-warehouse/ (pristupljeno 20.3.2021).

[10] I. Abramson, "Data Warehouse: The Choice of Inmon versus Kimball," 2016, [Online]. Dostupno na: https://vdocuments.site/inmon-vs-kimball-1.html. (pristupljeno 01.03.2021).

[11] T. Naeem, "Data Warehouse Concepts: Kimball vs. Inmon Approach," Astera, 2020. Dostupno na: https://www.astera.com/type/blog/data-warehouse-concepts/ (pristupljeno 01.03. 2021).

[12] D. Moody and M. A. R. Kortink, "From enterprise models to dimensional models: a methodology for data warehouse and data mart design," DMDW, 2000, [Online]. Dostupno na: http://ceur-ws.org/Vol-28/paper5.pdf. (pristupljeno 01.02.2021).

[13] "Create Star Schema Data Model in SQL Server with Microsoft Toolset." Dostupno na: https://www.mssqltips.com/sqlservertip/5690/create-a-star-schema-data-model-in-sql-server-using-the-microsoft-toolset/ (pristupljeno 10.4.2021).

[14] T. B. Pedersen, “Multidimensional Modeling,” in Encyclopedia of Database Systems, L. Liu and M. T. Özsu, Eds. Boston, MA: Springer US, 2009, pp. 1777–1784.

[15] M. Russo and A. Ferrari, Tabular modeling in microsoft SQL server analysis services, Second edition. Redmond, Washington: Microsoft Press, 2017.

[16] “What is OLAP? Cube, Operations & Types in Data Warehouse.” (bez dat.). Dostupno na: https://www.guru99.com/online-analytical-processing.html (pristupljeno 15.03.2021).

[17] T. Gryshko, “Multidimensional vs Tabular SSAS models | Flexmonster,” Oct. 15, 2020. https://www.flexmonster.com/blog/multidimensional-vs-tabular/ (pristupljeno 11.04.2021).

[18] R. Srinivasulu, “Comparing SSAS Tabular and Multidimensional Models | Pluralsight,” Jul. 17, 2019. https://www.pluralsight.com/guides/comparing-ssas-tabular-and-multidimensional-models (pristupljeno 11.04.2021).

[19] M. Novak, D. Kermek, and I. Magdalenić, “Prijedlog arhitekture za generator opisnika ETL procesa,” presented at the Central European Conference on Information and Intelligent system, 2019, [Online]. Dostupno na: http://archive.ceciis.foi.hr/app/public/conferences/2019/Proceedings/Croatian/CECIIS-2019_paper_32-HR.pdf. (pristupljeno 15.03.2021).

[20] V. Gour, S. S. Sarangdevot, G. Singh Tanwar, and A. Sharma, “Improve Performance of Extract, Transform and Load (ETL) in Data Warehouse,” International Journal on Computer Science and Engineering, vol. 02, no. 03, pp. 786–789, 2010.

[21] S. Sajida and S. Ramakrishna, “A Study of Extract-Transform-Load (ETL) Processes,” International Journal of Engineering Research & Technology, vol. 3, no. 18, pp. 1–6, 2015.

[22] „ETL Process Overview — ETL Database“, Stitch. (bez dat.). Dostupno na: https://www.stitchdata.com/etldatabase/etl-process/ (pristupljeno 06.04.2021).

[23] „ETL Transform — ETL Database“, Stitch. (bez dat.). Dostupno na: https://www.stitchdata.com/etldatabase/etl-transform/ (pristupljeno 27.03.2021).

[24] „ETL (Extract, Transform, and Load) Process in Data Warehouse“ (bez dat.). Dostupno na: https://www.guru99.com/etl-extract-load-process.html (pristupljeno 27.03.2021).

[25] “Developer tools, technical documentation and coding examples.” Dostupno na: https://docs.microsoft.com/en-us/ (pristupljeno 22.4.2021).

[26] “Microsoft SQL samples - SQL Server.” Dostupno nahttps://docs.microsoft.com/en-us/sql/samples/sql-samples-where-are (pristupljeno 12.4.2021).

[27] “SQL Server technical documentation - SQL Server.” Dostupno na: https://docs.microsoft.com/en-us/sql/sql-server/ (pristupljeno 14.4.2021).

[28] “Date Dimension File.” Dostupno na: https://support.sisense.com/kb/en/article/date-dimension-file (pristupljeno 01.04.2021).

IZJAVA

Izjavljujem pod punom moralnom odgovornošću da sam diplomski rad izradio samostalno, isključivo znanjem stečenim na Odjelu za elektrotehniku i računarstvo, služeći se navedenim izvorima podataka i uz stručno vodstvo mentora izv. prof. dr. sc. Marija Miličevića i komentorice Ines Obradović, mag. ing. comp., kojima se još jednom srdačno zahvaljujem.

Vlaho Stanović