Platforma Microsoft Azure

Kepe, Martin

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Martin Kepe

Integracija podataka pomoću Azure servisa

DIPLOMSKI RAD

Varaždin, 2023.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Martin Kepe

Matični broj: 44017/15-R

Studij: Informacijsko i programsko inženjerstvo

Integracija podataka pomoću Azure servisa

DIPLOMSKI RAD

Mentor:

Prof. dr. sc. Kornelije Rabuzin

Varaždin, travanj 2023.

Martin Kepe

Izjava o izvornosti

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

_____

Sažetak

Ovaj rad teorijski i praktično obrađuje integraciju podataka u skladište podataka. Kroz teorijski dio obuhvaćene su teme koje je potrebno razumjeti prije same integracije unutar skladišta podataka, a to su: ETL, činjenične i dimenzijske tablice i sporo mijenjajuće dimenzije.

Drugi dio rada se odnosi na praktični dio gdje se na primjeru odrađuju sve spomenute teorijske teme. Primjer projekta je rađen za zamišljeno poduzeće koje se bavi prodajom. Podatci se prikupljaju iz operacijske baze podataka i učitavaju u dimenzije i činjenicu. Prije učitavanja podataka potrebno je podatke pripremiti. Na kraju je cilj imati implementirano potpuno funkcionalno skladište podataka, te implementirane cjevovode (engl. pipelines) koji će inicijalno napuniti skladište podataka, ali i automatski osvježavati skladište podataka s novim podatcima. Nakon što se podatci učitaju u skladište podataka, prikazani su grafovi koji daju neke korisne informacije. Na taj način se želi dočarati svrha integracije podataka u skladište podataka.

Ključne riječi: ETL; Azure; Data Factory; Skladište podataka; PowerBI

Sadržaj

1. Uvod

U današnje vrijeme prikupljanje podataka je potpuno normalan, ali i nužan proces za sve tvrtke koje žele ostati konkurentne na tržištu. Prije nekoliko godina nije bilo moguće spremati velike količine podataka te ih procesirati. Tako da su se informacije iz podataka dobivale pomoću slučajno odabranog uzorka, dobivene informacije su se onda odnosile na cijeli skup. Korištenje uzorka nije savršena tehnika, ali je bila jedina moguća u tom trenutku. S vremenom je postajalo moguće prikupljati ogroman broj podataka uz prihvatljive troškove te su i dobivene informacije iz podataka bile kvalitetnije. Trend prikupljanja ogromne količine podataka je sve prisutan, ali samo prikupljanje i spremanje podataka je daleko od gotovog posla ukoliko je cilj pomoću podataka ostvariti neku prednost. Velike količine podataka, koje se svakodnevno generiraju iz različitih izvora predstavljaju izazov za organizacije koje žele iskoristiti njihov potencijal. Da bi se podaci iskoristili, potrebno ih je strukturirati, čistiti i smjestiti na pravo mjesto u pravo vrijeme. Skladište podataka je sustav koji se koristi za skupljanje, pohranjivanje i analizu podataka iz različitih izvora, a integracija podataka u njega je ključni korak u procesu upravljanja poslovnim informacijama. Skladište podataka zahtjeva da podatci budu strukturirani prije njihove pohrane za razliku od nekih drugih opcija poput podatkovnih jezera.

Automatizacija procesa prikupljanja, čišćenja i učitavanja podataka je nužna zbog velike količine podataka. Integracija podataka kroz oblak je vrlo popularna, jer tvrtke više ne moraju izdvojiti velika davanja za hardver, već se sve može iznajmiti preko interneta. Microsoft Azure je platforma koju nudi razvoj i upravljanje različitim aplikacijama i uslugama u oblaku, a pruža različite servise za integraciju podataka u skladište podataka.

Cilj ovog rada je dati pregled dostupnih Azure servisa za integraciju podataka te pokazati kako se oni mogu koristiti za automatizaciju i poboljšanje procesa integracije podataka u skladište podataka. U prvom dijelu rada teorijski su opisani svi pojmovi koje je potrebno razumjeti kako bi se mogao pratiti praktični dio. Praktični dio se odnosi na trgovinu koja se bavi prodajom, cilj je prebaciti podatke iz transakcijske baze podataka u skladište podataka. Kako bi se dobio uvid u vrijednosti koje podatci mogu dati, napravljeni su grafovi na temelju podataka koji su integrirani u skladište podataka. Grafovi daju interaktivan uvid u poslovanje te načine na koje se ono može unaprijediti

2. Metode i tehnike rada

Metoda istraživanja koja je korištena za ovaj rad obuhvatila je dva ključna aspekta: pregled književne građe u biblioteci i pretraživanje interneta pomoću *Google Scholar* tražilice.

Kako bi se izvršio projektni dio rada, korišteni su Microsoft Azure servisi, uključujući SQL server, *Blob Storage* i *Azure Data Factory.* SQL Server Management Studio (SSMS) korišten je za pristupanje SQL serveru i pisanje SQL koda, dok je Microsoftov alat *PowerBI* korišten za vizualizaciju podataka.

U okviru diplomskog rada, za sastavljanje teksta korišten je Microsoft Word, a za navođenje literature korišten je program *Zotero*.

3. Skladište podataka

Skladišta podataka se pojavljuju između 1980. i 1990. godine. Njihova uloga je riješiti sve veću potražnju visokog menadžmenta za analitičkim izvještajima. Normalne transakcijske baze podataka nisu efikasno zadovoljavale analitičke potrebe (Chaudhary et al., 2011).

Operativne baze podataka nisu napravljene kako bi odgovarale na analitičke upite već da odgovaraju na operativne upite. Operativni upiti su vrlo jednostavni upiti koji se postavljaju velikom brzinom, te se odgovori očekuju u kratkom vremenu. Analitički upiti mogu biti vrlo složeni te se na njihov odgovor troši mnogo resursa, kako bi se doskočilo tom problemu napravljena su skladišta podataka koja su služila za odgovaranje na analitičke upite.

Bill Inmon kojeg smatramo kao jednog od oca skladišta podataka, definirao je skladište podataka kao subjektno orijentirano, integrirano, različito u vremenu i nepromjenjivo prikupljanje podataka u svrhu donošenja odluka od strane menadžera (Inmon, 2002).

Kako bi se bolje razumjela definicija, u nastavku su objašnjeni pojedini atributi koje je Inmon koristio kod definiranja skladišta podataka.

Subjektno orijentirano – podaci su organizirani po temama kojima korisnik želi pristupati. Dok su transakcijske baze podataka obično oblikovane oko procesa, skladište podataka je oblikovano oko subjekta npr. prodaja. Integrirano – Podaci se „čiste“. Uklanjaju se neispravni i nepotpuni podatci koji se ne mogu iskoristiti. Svi podatci jednoznačno su određeni nakon integracije. Nepromjenjivo – korisnici samo pristupaju podatcima, ne mogu ih ažurirati kao što je obično slučaj kod transakcijskih baza podataka, ne mogu ni dodavati nove podatke u stvarnom vremenu. Podatci se dodaju u točno određenom vremenu koje je definirano, s obzirom na poslovanje. Obično je kod integracije podataka skladište podataka nedostupno. Različito u vremenu – moguće je gledati podatke kroz vrijeme. Skladište podataka čuva povijest.

Zahtjevi koje skladište podataka treba ispunjavati prema (Kimball & Ross, 2013) su:

- Skladište podataka mora omogućiti lagani pristup informacijama (jednostavnost i brzina): Informacije trebaju biti razumljive i intuitivne krajnjem poslovnom korisniku. Alati za poslovnu inteligenciju koji pristupaju podatcima trebaju biti jednostavni krajnjem korisniku. Podatcima se pristupa sa što manjim vremenom čekanja.
- Skladište podataka prikazuje informacije konzistentno: Podatci moraju biti točni kako bi se došlo do točnih analitičkih informacija. Prije korištenja podataka od strane krajnjih korisnika, podatke je potrebno očistiti i pripremiti.

- Skladište podataka se mora prilagoditi promjenama: Skladište podataka se radi kako bi se odgovorilo zahtjevima poslovanja u budućnosti te spada pod strateški cilj. U tom vremenu mogu se dogoditi novi zahtjevi poslovnih korisnika na koje je potrebno odgovoriti što efikasnije. Aplikacije koje su povezane uz skladište podataka mogu zastarjeti ili mogu izaći bolja rješenja. Iz tog razloga potrebno je graditi skladište podataka koje će moći odgovoriti na promjene u budućnosti. Također moguća je i promjena podataka na izvoru, te promjene je potrebno i zabilježiti u skladištu podataka na način kako je to definirano u zahtjevima klijenta.
- Skladište podataka prikazuje informacije u određenom vremenu: Ukoliko se informacije iz skladišta podataka koriste u operacijskim odlukama potrebno je biti svjestan vremena koje je potrebno da se podatci očiste i transformiraju kako bi se mogli i iskoristiti.
- Skladište podataka mora imati visok stupanj sigurnosti: Skladište podataka sadrži velik broj podataka koje su vrijedne organizaciji koja ih stvara. Izloženost tih podataka bi ugrozila stajalište organizacije naspram svoje konkurencije. Sigurnost je vrlo bitna za organizacije.
- Skladište podataka mora služiti kao potpora odlučivanju. Svrha skladišta podataka je omogućavanje i olakšavanje donošenja pravih i pravovremenih odluka. Tako da odluke koje se donose na temelju analize podataka moraju biti usmjerene k unaprjeđenju poslovanja.
- Skladište podatka se mora koristiti: Ukoliko se i proizvede vrlo dobro rješenje, ukoliko se ne koristi, projekt izrade skladišta podataka je neuspješan. Skladište podataka nije svrha samom sebi, potrebno je izgraditi nešto što će krajnji korisnici koristiti i imati koristi od toga.

Tablice u skladištu podataka se dijele na dimenzijske i činjenične. Skladište podataka je sustav koji izvlači, čisti, priprema i isporučuje podatke s izvora u dimenzijsku podatkovnu struktura te pruža i implementira postavljanje upita nad podatcima te njihovu analizu u svrhu pružanja podrške kod donošenja odluka (Kimball & Casetra, 2004).

Dimenzijski model omogućuje brže odgovaranje na postavljene upite od strane korisnika tj. zahtjeva manje resursa. Također dimenzijski model je otporniji na promjene i prezentira podatke krajnjim korisnicima na razumljiv način. Prednosti tog modela u odnosu na druge se povećava kako se i povećava količina podataka.

3.1. Činjenične tablice

Činjenične tablice sadrže podatke koji se odnose na mjere i kontekstualne podatke. Svaki podatak u činjeničnoj tablici obično predstavlja poslovnu transakciju, događaj bitan za poslovanje ili poslovni proizvod (Patel & Patel, 2012).

Mjere u činjeničnoj tablici su brojčane vrijednosti koje predstavljaju stvari poput iznosa transakcije, cijene proizvoda, današnje temperature itd. Moguće je da mjere budu i tekstualne u rijetkim slučajevima, ali takvi atributi se nastoje izbjeći te prebaciti u dimenzije. Osim mjera u činjeničnoj tablici se nalaze i kontekstualni podatci tj. vanjski ključevi na tablice koje daju kontekst mjerama. Tablice koje činjenične referenciraju sa svojim vanjskim ključevima nazivamo dimenzijskim tablicama.

3.2. Dimenzijske tablice

Dimenzijske tablice sadrže atribute koje opisuju mjere unutar činjeničnih tablica. Dimenzije daju odgovore na pitanja poput: Tko?, Što?, Kad?, Gdje?, Kako? i Zašto? (Kimball & Ross, 2013).

Uz tablicu činjenica i svih pripadajućih dimenzija dolazi se do potpunih podataka za jedan unos u skladište podataka. Opisni atributi koje sadrži dimenzija mogu biti i brojčani i tekstualni, obično ih je puno. Činjenične tablice referenciraju dimenzijske uz pomoć vanjskih surogat ključeva. Dimenzije sadrže prirodni ključ i surogat ključ.

Poslovni (prirodni) ključ se sastoji od jednog ili više polja sa samog izvora podataka. Na taj način se zadržava veza s izvorom podataka te se mogu ažurirati podatci u skladištu podataka ukoliko se promjene na izvoru i ukoliko je to zahtjev koji skladište podataka mora ispunjavati. Više o promjenama nad dimenzijama spominje se u nastavku ovog rada.

Surogat ključ je ključ koji se generira za svaki zapis kod njegovog unos u skladište podataka. Na taj način se osigurava da svaki zapis u skladištu podataka ima jedinstveni ključ te se pomoću njega jedinstveno identificira.

3.3. Sporo mijenjajuće dimenzije

Kod unošenja novih podataka u skladište podataka, prvo se provjerava postoji li već isti podatak u skladištu. Ta provjera se vrši koristeći poslovni ključ jer se on nalazi na izvoru i u dimenziji, što omogućava lakšu identifikaciju istih podataka. Tijekom provjere podataka prije integracije, moguća su tri scenarija: podatak ne postoji u skladištu i treba ga unijeti, podatak

već postoji u skladištu i svi atributi su potpuno jednaki, u tom slučaju taj redak se preskače, i treći slučaj kada podatak već postoji u skladištu, ali su neki atributi različiti (Santos & Belo, 2011). Treći slučaj je onaj kod kojeg se koriste sporo mijenjajuće dimenzije. Postoji nekoliko tipova sporo mijenjajućih dimenzije. Svaki tip ima svoja pravila za bilježenje promjena.

Na svim tipovima sporomijenjajućih dimenzija (SCD) koji su opisani u nastavku korišten je primjer promjene *smjera* s *Informacijskih sustava* na *Poslovne sustave*. SID kratica predstavlja surogat ključ, a BID kratica predstavlja poslovni ključ.

Kod tipa 1 sporo mijenjajuće dimenzije ažurira se vrijednost u skladištu podataka. Ovaj način omogućuje ispravak pogrešno unesenih podataka ili ažuriranje atributa u podatcima koji nisu bili poznati prije. Povijest se ne prati tako da se ovaj tip koristi samo kada praćenje promjena nije bitno.

Prije:

Tablica 1: Originalan podatak - SCD1

SID	BID	Naziv	Lokacija	Smjer
1	1	FOI	Varaždin	Informacijski sustavi

Poslije:

Tablica 2: Ažuriran podatak - SCD1

SID	BID	Naziv	Lokacija	Smjer
1	1	FOI	Varaždin	**Poslovni sustavi**

Kod tipa 2 se dodaje novi redak u skladište podataka. Ovaj tip se koristi kada je bitno pratiti povijest i promjene atributa tokom vremena. Moguće je i dodavanje dodatnih datumskih kolona koje označavaju vrijeme od kada do kada je određeni podatak vrijedio. Osim datumskih kolona moguće je dodati zastavicu koja određuje trenutno vrijedeći tj. ne vrijedeći podatak.

Prije:

Tablica 3: Originalan podatak - SCD2

SID	BID	Naziv	Lokacija	Smjer	Aktivan	Vrijedi od:	Vrijedi do:
1	1	FOI	Varaždin	Informacijski sustavi	DA	1.1.2021.	1.1.9999.

Poslije:

Tablica 4: Ažuriran podatak - SCD2

SID	BID	Naziv	Lokacija	Smjer	Aktivan	Vrijedi od:	Vrijedi do:
1	1	FOI	Varaždin	Informacijski sustavi	Ne	1.1.2021.	2.2.2021.
2	1	FOI	Varaždin	**Poslovni sustavi**	Da	2.2.2021.	1.1.9999.

Na prikazanom primjeru promjena smjera se dogodila 2.2.2021, to se vidi po datumskim kolonama. Novi redak koji je i trenutno aktivan ima isti poslovni ključ kao i prijašnji vrijedeći slog za taj podatak. Pomoću poslovnog ključa moguće je doći do cijele povijesti i promjena nad jednim zapisom. Ovaj tip je dobar ukoliko se ne očekuje velik broj promjena jer se svakom novom promjenom upisuje novi slog što će znatno povećavati dimenziju ukoliko su promjene česte.

Kod tipa 3 se dodaje nova kolona koja će sadržavati prošlu vrijednost atributa, a trenutna vrijednost se ažurira. Ovaj tip je limitiran na broj promjena, jer se čuva samo prošla vrijednost atributa. Ukoliko je potrebno spremati sve promjene atributa ovaj tip to ne omogućuje. Moguće je dodavati novu kolonu za svaku promjenu atributa, ali to je loša praksa. Također moguće je spremati staru vrijednost samo za jedan atribut, svako spremanje stare vrijednosti određenog atributa zahtjeva dodatnu kolonu.

Prije:

Tablica 5: Originalan podatak - SCD3

SID	BID	Naziv	Lokacija	Smjer	Prijašnji smjer
1	1	FOI	Varaždin	Informacijski sustavi	_

Poslije:

Tablica 6: Ažuriran podatak - SCD3

SID	BID	Naziv	Lokacija	Smjer	Prijašnji smjer
1	1	FOI	Varaždin	**Poslovni sustavi**	Informacijski sustavi

Vrlo jednostavan tip sporo mijenjajućih dimenzija gdje se samo dodaje nova kolona koja sadrži prošlu vrijednost smjera. Čuva se samo prošla vrijednost atributa, ne cijela povijest.

Kod tipa 4 izdvaja se atribut koji se često mijenjaju unutar dimenzija u posebnu mini dimenziju. Koristeći SCD 2 za čuvanje povijesti svaka promjena atributa znači i novi redak unutar dimenzije. Kako bi se izbjeglo brzo punjenje osnovne dimenzije koristi se tip 4 kod kojeg se sve promjene učestalo mijenjanih atributa zapisuju u novu mini dimenziju. Obje dimenzije su posebno povezane s činjeničnom tablicom vanjskim ključevima, ali dimenzije nisu međusobno povezane.

Kod tipa 5 izdvaja se atribut koji se često mijenjaju unutar dimenzija u posebnu dimenziju koja će uz atribute sadržavati i vanjski ključ na originalnu dimenziju. Na taj način je moguće povezati te dvije dimenzije i u prikazu ih koristiti kao cjelinu bez povezivanja preko činjenične tablice. Za razliku od tipa 4 moguće je povezati mini dimenziju s dimenzijom iz koje je izdvojena.

Kod tipa 6 se dodaje novi atribut u tablicu koji će sadržavati prijašnje stanje atributa. Uz to dodaju se datumske kolone koje prikazuju vrijeme od kada do kada je vrijedio određeni podatak, poželjno je i dodati zastavicu koja označuje trenutno vrijedeći podataka. Tip 6 kombinira nekoliko prijašnje opisanih tipova: SCD 1- ažuriranje atributa koji se promijenio nad svim redcima sa istim poslovnim ključem, SCD 2 – dodavanje datumskih kolona i zastavice, SCD 3 dodavanje kolone koja čuva prošlu vrijednost atributa za određeni vremenski period.

Tablica 7: Ažuriran podatak – SCD6

SID	BID	Naziv	Lokacija	Smjer	Prijašnji smjer	Aktivan	Vrijedi od:	Vrijedi do:
1	1	FOI	Varaždin	**Poslovni sustavi**	Informacijski sustavi	Ne	1.1.2021.	2.2.2021.
2	1	FOI	Varaždin	**Poslovni sustavi**	Poslovni sustavi	Da	2.2.2021.	1.1.9999.

Na prikazanom primjeru promjena smjera se dogodila 2.2.2021. Dodan je novi redak, ažurirana informacija prijašnjeg smjera i ažurirana je informacija trenutnog smjera.

3.4. Shema zvijezde

Najčešće korištena arhitektura kod gradnje skladišta podataka je zvijezda shema. Činjenična tablica nalazi se u sredini ovog modela te se dimenzijske tablice nalaze okolo činjenične. Dimenzije nisu međusobno povezane, povezane su samo s činjeničnom tablicom. Struktura koja podsjeća na zvijezdu. Model je jednostavan i ima dobre rezultate što se tiče brzine izvođenja pa je zato i najpopularniji.

Denormalizacija tablica u transakcijskoj bazi podataka dovodi do određene razine redundancije podataka. U budućnosti ta redundancija pridonosi brzini skladišta podataka jer do podataka nije potrebno dolaziti preko većeg broja tablica.

3.5. Pahuljasta shema

Kod pahuljaste arhitekture skladišta podataka činjenična tablica je u sredini tj. sadrži vanjske ključeve dimenzijskih tablica, ali dimenzijske tablice nisu denormalizirane.

Kod normalizacije dimenzija one se razdvajaju u veći broj dimenzija te su također povezane vanjskim ključevima. Tako da su dimenzije raspoređene po razinama. Razine tj. broj dimenzija se povećavaju s obzirom na broj normaliziranja koja se primjene na dimenzijske tablice. Na taj način sama arhitektura podsjeća na pahulju pa od tud i ime (Sarka et al., 2012).

Za razliku od sheme zvijezde, kod pahuljaste sheme nisu sve dimenzije direktno povezane s činjeničnom tablicom. Zbog toga je potrebno kod postavljanja upita odraditi veći broj spajanja tablica kako bi se došlo do određenih informacija od dimenzija koje su povezane samo s drugim dimenzijama.

Broj dimenzija povezanih na činjeničnu tablicu predstavlja razinu zrnatosti do koje se može doći analizom (Sarka et al., 2012).

4. Integracija podataka

Danas nije neobično za poduzeća da imaju više izvora podataka iz više različitih sustava. Kako bi se iskoristio puni potencijal podataka potrebno ih je čuvati na jednom mjestu jer se iz većeg broja podataka dolazi do kvalitetnijih rezultata na analitičke upite. Nije neobično da struktura podataka bude različita na svakom posebnom sustavu. To je jedan od izazova integracije podataka kojeg je potrebno riješiti prije učitavanja podataka.

Cilj integriranja podataka u sustav je omogućiti univerzalni pristup do podataka koji dolaze iz izvora podataka. Sustav preko kojeg se pristupa podatcima je autonoman i heterogen (Doan et al., 2012).

Univerzalni pristup podrazumijeva pristup podatcima koji i ne moraju dolaziti s istog izvora, ali im je moguće pristupiti s jednog mjesta neovisno o broju sustava s kojih pristižu podatci.

Heterogeni sustav podrazumijeva pristupanje podatcima neovisno o njihovom formatu. Podatci mogu dolaziti iz baze podataka, ali mogu dolaziti i u obliku tekstualnih datoteka ili nekih drugih oblika i mogu se pregledavati na jednom mjestu bez obzira na format.

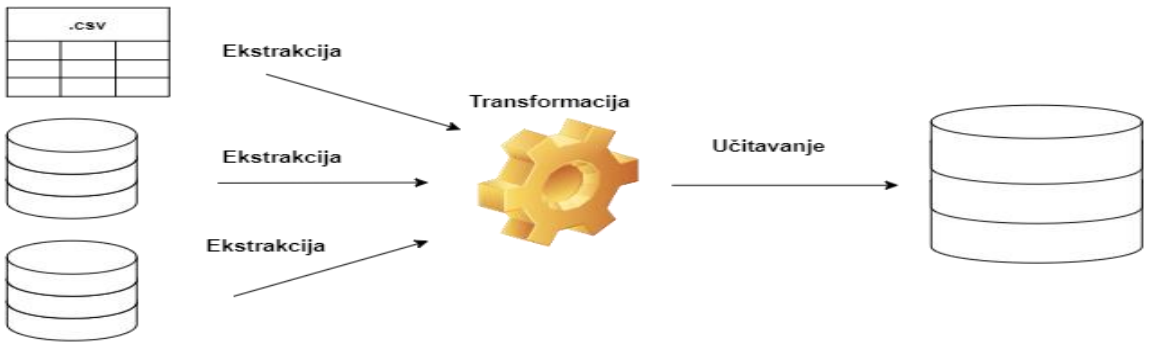
Izvori podataka se mogu nalaziti na više različitih sustava. Svaki taj sustav može imati svoje restrikcije kod pristupa do podataka. Autonomnost sustava predstavlja pristup do svih tih podataka s jednog mjesta bez obzira na restrikcije pojedinih izvora podataka.

Postoji više tipova i tehnika integracije podataka, počeci integracije podataka su se odnosili na ručno ukucavanje podataka u sustav, danas su ti procesi automatizirani. Neki od tipova integracije podataka su ETL, ELT, strujanje podataka (engl. data streaming) i virtualizacija podataka (engl. data virtualization). U ovom radu najveći fokus će biti na integraciji podataka u skladište podataka koristeći ETL postupak. Automatizacijom integracije podataka u skladište podataka skraćuje se vrijeme koje potrebno kako bi se došlo do potrebnih informacija na temelju analize podataka.

Zaključno, integracija podataka je ključni aspekt modernih poduzeća. Omogućuje organizacijama da donose informirane odluke, stječu vrijedni uvid u podatke i poboljšaju učinkovitost svojih operacija. Pažljivo razmatrajući čimbenike uključene u integraciju podataka, organizacije mogu osigurati učinkovitu i sigurnu integraciju te postići puni potencijal svojih podataka.

4.1. ETL

ETL je akronim koji dolazi od riječi: ekstrakcija, transformacija i učitavanje (engl. extract, transform and load). Koristi se kod procesa integracija podataka. Kao što i proizlazi iz akronima odnosi se na ekstrakciju, transformaciju i učitavanje podataka u odredište. Pojednostavljeni prikaz ETL procesa je prikazan na donjoj slici.



Slika 1: ETL proces

Na gornjoj slici prikazan je proces kod prebacivanja podataka iz izvora u odredište. Radi se o kompleksnoj kombinaciji procesa i tehnologija na koji se troši značajna količina resursa. Zahtjeva vještine poslovnih analitičara, arhitekta rješenja i programera (El-Sappagh et al., 2011).

Jedan od zadatka poslovnog analitičara je definirati opseg rješenja. Pod opsegom rješenja misli se na podatke koje je potrebno obuhvatiti kod izvlačenja podataka, te kako bi oni trebali izgledati na odredištu. Arhitekt rješenja radi na izgledu cjelokupnog rješenja ETL procesa. Programer je zadužen za automatiziranje cijelog procesa u nekom od ETL alata.

ETL Proces podijeljen je na tri faze, a to su:

- Ekstrakcija - Dohvaćanje podataka iz više različitih izvora koji mogu biti: baze podataka, tekstualni dokumenti, CRM sustavi itd. Podatci mogu biti vrlo različite strukture s različitim ograničenjima te ih je potrebno učitati u jedinstvenu bazu podataka. Kako bi to bilo moguće potrebno je detaljno analizirati podatke s izvora.
- Transformacija - Transformiranjem i čišćenjem podataka želi se postići bolja kvaliteta podataka. Opseg transformacija koje je potrebno izvršiti ovisi o kvaliteti podataka na izvoru, ali i o poslovnim zahtjevima korisnika. Transformacije koje je potrebno odraditi nad samim podatcima mogu obuhvaćati (Kimball & Casetra, 2004):

- Eliminacija atributa, objekata ili polja koji ne ulaze u opseg rješenja tj. ne doprinose kvaliteti završnog proizvoda ili rješenja, uzimajući u obzir zahtjeve korisnika.
- Postavljanje zastavica kojima se ukazuje na podatke koji nedostaju i generiranje specijalnih surogat ključeva.
- Dopunjavanje podataka vrijednostima koje izgledaju kao najbolja zamjena u slučaju kada su podatci oštećeni.
- Ljudska intervencija na razini retka npr. smanjivanje duljine niza znakova (engl. string) koji prekoračuju maksimalnu moguću. Nizu znaka se izbacuju dijelovi koji su suvišni, ako je to moguće.
- Razvoj normalizirane reprezentacije podataka.

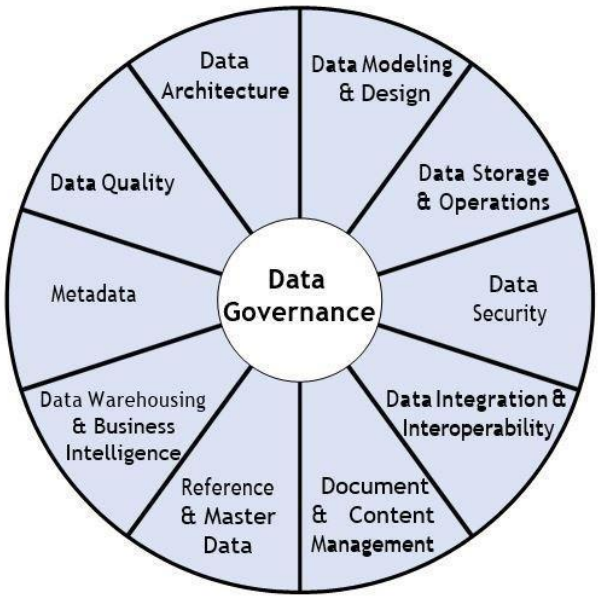
- Učitavanje - Učitavanje podataka u samo odredište. Učitani podatci su oni kojima će pristupati krajnji korisnici. Prema (Arul Kumar et al., 2019) poznajemo tri vrste učitavanja:
  - Inicijalno – obuhvaća učitavanje podataka u sve tablice koje ulaze u opseg rješenja. Prije učitavanja sve tablice na odredištu su prazne.
  - Inkrementalno – nakon inicijalnog učitavanja pokreće se inkrementalni u određenim vremenskim razmacima ili kod određenog događaja. Obično su tu podatci koji su kreirani nakon inicijalnog učitavanja ili su promijenjeni nakon inicijalnog učitavanja.
  - Potpuno osvježavanje – nad jednom ili više tablica se izvodi potpuno brisanje podataka te se ponovno učitavaju podatci.

4.2. Upravljanje podacima

Upravljanje podacima (engl. data governance) je disciplina koja se često zanemaruje u poslovanju. Učinkovito upravljanje podatcima omogućuje organizaciji iskorištavanje punog potencijala prikupljenih podataka uz što manji trošak resursa.

Upravljanje podacima kao disciplina predstavlja organizaciju i implementaciju: procedura, politike, strukture, uloga i odgovornosti koji definiraju pravila za efektivno korištenje podataka (Ladley, 2019). Upravljanje podacima određuje pravila rukovanja s podatcima koje menadžment prati kod rada s podatcima (Cheong & Chang, 2007).

Poduzeća moraju konstantno raditi na svojim tehnikama upravljanja podataka. Informacijski svijet se konstantno mijenja, kako bi poduzeća bila u korak s promjenama potrebno je provjeravati, procjenjivati i evaluirati tehnike upravljanja podatcima. Ljudi i tehnologije su oni koji trebaju pratiti te promjene i implementirati ih unutar poduzeća.



Slika 2: DAMA Kotač (Izvor: DAMA International)

Dama kotač koji se nalazi na slici prikazuje važnost discipline upravljanja podataka (engl. data governance). U samom srcu kotača nalazi se upravljanje podatcima, ostale discipline koje se odnose na rad s podacima se nalaze na rubovima. To nije slučajno, na taj način se želi prikazati važnost upravljanja podacima te sve discipline koje spadaju pod upravljanje podataka.

4.3. Kvaliteta podataka

Kvaliteta podataka se često spominje u ovom radu jer je vrlo važna kada je riječ o podatcima. Upravljanje podacima poboljšava kvalitetu podataka pa će se u ovom pod poglavlju detaljnije opisati sama kvaliteta podataka.

Prema (Cai & Zhu, 2015) elementi kvalitete podataka su :

- Pristupačnost – razina teškoće pristupa podatcima.
- Pravovremenost – podatci su dostupni korisnicima u pravo vrijeme. Vrijeme od nastanka podataka, pa do njegove integracije i na kraju samog korištenja trebalo bi biti što kraće kako bi podatci i dalje bili iskoristivi.
- Autorizacija – odnosi se na pravnu stranu korištenja podataka. Organizacija mora imati mogućnost legalnog korištenja nastalih podataka.
- Dokumentiranost – podatci se mogu naći u specifikacijama gdje je jasno definirana njihova uloga, definicija, formati, poslovna pravila i drugo.

- Meta podatci – daje informacije o samim podatcima. O meta podatcima će biti više riječi u posebnom poglavlju u ovom radu.
- Točnost – podaci prikazuju stvarne rezultate proizašle iz poslovnih procesa.
- Konzistentnost – isti podatci imaju isti prikaz bez obzira nalaze li se na drugačijim sustavima.
- Integritet – podatci se ne mijenjaju bez potrebnih ovlaštenja.
- Potpunost – svi podatci sadrže sve potrebne vrijednosti.

Kvaliteta podataka se može osiguravati na nekoliko mjesta ako pričamo o skladištu podataka. Prva razina osiguravanja kvalitete podataka se može implementirati kod samog ulaza podataka. Aplikacije koje generiraju podatke kroz unos korisnika se mogu optimizirati raznim provjerama u smislu kvalitete podataka. Druga razina je osiguravanje kvalitete podataka prije učitavanja u skladište podataka. Prije učitavanja podataka u samo skladište podataka, podatci se učitavaju u pripremnu bazu podataka gdje se izvode razne transformacije i čišćenja kako bi podatci bili što kvalitetniji. Treća razina je osiguravanje kvalitete podataka u skladištu podataka. Čuvanje povijesnih podataka može biti vrlo važno za određene podatke, kako se u skladištu podataka ne bi nalazili zastarjeli podatci ili dupli podatci koriste se dimenzije koje se sporo mijenjaju.

5. Tehnologije na oblaku

Ideja o oblaku (*engl. cloud*) na kojem se nalaze resursi te kojima može svatko pristupiti se pojavljivala već 1960-ih godina. John McCarthy je predložio sustav dijeljenja vremena, što bi omogućilo pristupanje više korisnika nad jednim dijeljenim računalom (de Bruin & Floridi, 2017). Takav koncept predstavlja pojednostavljeni način na koji oblak i dijeljenje resursa funkcionira danas.

Tehnologije na oblaku su revolucionizirale načine na koji se spremaju, pristupaju i dijele informacije. Računarstvo na oblaku (*engl. cloud computing*) predstavlja resurse poput softvera, pohrane i procesiranja kojima se pristupa preko interneta. Korištenje tehnologija na oblaku koriste tvrtke jer im omogućuje uštedu, fleksibilnost, skalabilnost i sigurnost.

Ušteda se ostvaruje jer se ne moraju izdvajati velike svote novca kako bi se kupio sav potreban hardver. Uz hardver potrebno je imati i stručno osoblje koje zna raditi s njime, a to se odnosi na postavljanje i održavanje. Pristupanje resursima preko oblaka je vrlo fleksibilno. Potrebna je internetska veza koja je danas dostupna gotovo svugdje i uređaj s kojeg se pristupa što može biti pametni telefon, tablet, računalo i drugi uređaji s kojima je moguće pristupiti internetu. Resursi na oblaku se koriste po potrebi, što znači da je moguće po potrebi smanjivati i povećavati količinu korištenih resursa. Na taj način se smanjuje rizik od nedostatka dostupnih resursa. Sigurnost na oblaku je na vrlo visokoj razini. Pružatelj usluga je zadužen za sigurnost te se ona shvaća vrlo ozbiljno. Koriste se razne enkripcije i vatrozidovi kako bi se zaštitilo od potencijalnih prijetnji.

Postoje tri glavna modela kada se govori o pružanju usluga preko oblaka, a to su: softver kao servis (SaaS), infrastruktura kao servis (IaaS) i platforma kao servis (PaaS).

SaaS je model kod kojeg se sama aplikacija nalazi na oblaku, a korisnici joj pristupaju preko interneta. Korisnici obično plaćaju određenu mjesečnu naknadu (Satyanarayana, 2011). Nije potrebna nikakva instalacija te se sam softver može prilagoditi za svakog korisnika, ali veće prilagodbe valja izbjegavati zbog kompleksnosti i uštede.

IaaS je model kod kojeg se preko oblaka uz korištenje interneta pristupa resursima hardvera i pridruženog softvera. Hardver kojem se pristupa pruža mogućnosti poput pohrane, servera i mreže. Pridruženi softver hardveru podrazumijeva operacijske i datotečne sustave (Bhardwaj et al., 2010). Na iznajmljeni hardver preko oblaka, korisnici podešavaju svoj softver tj. aplikacije po potrebi.

PaaS je model kod kojeg se pristupa hardveru i softveru preko oblaka. Na hardveru se nalaze i određene aplikacije koje se obično koriste kako bi olakšale razvoj i isporuku aplikacija

za inženjere (Bhardwaj et al., 2010). Ovaj model se i najčešće koristi kod razvoja aplikacija. Prednost je ta što se ne mora voditi briga o temeljnoj infrastrukturi te je to zadatak pružatelja usluga oblaka.

5.1. Pružatelji usluga na oblaku

Velike kompanije su vrlo brzo primijetile potrebu za pružanjem usluga na oblaku. Amazon je prvi predstavio Amazon Web Services (AWS) 2006. godine. AWS je trenutno i najpopularniji pružatelj usluga na oblaku s tržišnim udjelom od 34% u 2022. godini (*Infographic*, 2022). Postoje razni servisi koji se mogu naći na AWS-u, neki od onih koji se koriste pri radu s podatcima su:

- Amazon S3 – servis koji služi za pohranu objekata uz veliku skalabilnost, dostupnost, sigurnost i performanse. Amazon S3 se može koristiti za pohranu i zaštitu bilo koje količine podataka za različite svrhe, kao što su jezera podataka, web stranice, mobilne aplikacije, backup, arhive, poslovne aplikacije, IoT uređaji i analizu velikih podataka. Amazon S3 pruža značajke upravljanja tako da omogućuje optimiziranje, organizaciju i konfiguriranje pristupa podatcima kako bi se zadovoljili specifični poslovni i organizacijski zahtjevi (*What Is Amazon S3? - Amazon Simple Storage Service*, n.d.).
- AWS Glue – servis koji omogućuje otkrivanje, pripremu i integraciju podataka iz više izvora. Unutar alata vizualno se kreiraju ETL cjevovodi. Također, bez poslužitelja je, što znači da nema infrastrukture za upravljanje. Skalira se za bilo koju veličinu podataka i podržava sve vrste podataka i varijacije shema. Naplaćuje se po potrošnji što omogućuje veću agilnost i optimizaciju troškova(*What Is AWS Glue? - AWS Glue*, n.d.) .
- AWS Redshift – usluga skladištenja podataka u oblaku. Omogućuje organizacijama da pohrane i analiziraju velike količine podataka na vrlo skalabilan i ekonomičan način (*What Is Amazon Redshift?*, n.d.).

Microsoft Azure predstavljen je 2010. godine te 2022. godine posjeduje 21% tržišnog udjela u pružanju usluga preko oblaka (*Infographic*, 2022). Neki od servisa koji se koriste pri radu s podatcima su:

- Azure Data Factory (ADF) – servis koji omogućuje integraciju podataka kreiranjem cjevovoda. Cjevovodi koji se koriste za prijenos i transformaciju mogu se naštimati kako bi se izvršavali po točno određenom rasporedu. Podatci se mogu prenositi iz velikog broja raznih izvora u velik broj raznih odredišta. Cjevovodi se kreiraju korištenjem vizualnog sučelja što omogućuje vrlo jednostavnu izradu i praćenje njihovog

izvršavanja. Naplaćuje se po potrošnji, što omogućuje veliku agilnost i skalabilnost (*Azure Data Factory Documentation - Azure Data Factory*, n.d.).

- Azure Blob Storage – koristi se za pohranu podataka na oblaku. Omogućuje korisnicima pohranu i pristup velikim količinama nestrukturiranih podataka. Podatci koji se pohranjuju mogu biti videi, slike, tekstualni dokumenti i drugi (*Azure Blob Storage | Microsoft Azure*, n.d.).

Google Cloud predstavljen je 2008. godine. U 2022. godini nalazi se na trećem mjestu pružatelja usluga na oblaku s tržišnim udjelom od 11%. Servisi koji se koriste pri radu s podatcima su:

- Cloud Data Fusion – servis koji omogućuje transformaciju i obradu podataka u realnom vremenu. Pruža mogućnost stvaranja biblioteka koje sadrže kreirane transformacije i konekcije. Takve biblioteke moguće je dijeliti, validirati i ponovno koristiti. Kreiranje biblioteka poboljšava produktivnost podatkovnih inženjera uz dobru suradnju u timu (*Cloud Data Fusion*, n.d.).
- Cloud Storage – omogućuje skalabilni način pohrane objekata. Pohranjenim podatcima je moguće pristupiti iz bilo lokacije uz pristup internetu. Tipovi podataka koji se pohranjuju mogu biti slike, videi, audio datoteke, tekstualne datoteke i drugi (*Cloud Storage*, n.d.).

Amazon, Google i Microsoft su ozbiljno shvatili oblak i njegove mogućnosti. Sve tri firme ostvaruju velike prihode iznajmljivanjem resursa preko oblaka. U ovom radu su nabrojani samo neki servisi koje je moguće pronaći na oblaku, ali postoji ih mnogo više. Tako da je u današnje vrijeme potreba za fizičkim posjedovanjem resursa mala te se većina kompanija okreće ka korištenju resursa preko oblaka.

5.2. Skladišta podataka na oblaku

Skladišta podataka na oblaku omogućuju upravljanje podacima i pohranjivanje podataka u skladište podataka putem interneta. Ona pružaju organizacijama veću fleksibilnost, dostupnost, pouzdanost i uštedu u troškovima.

Fleksibilnost omogućuje organizacijama da koriste resurse po potrebi. Ovisno o trenutnim potrebama prilagođava se skladišni prostor i računalna snaga za obradu podataka. Dostupnost znači da je pristup podatcima moguć bilo gdje i bilo kada. Sve što je potrebno je pristup internetu. Pouzdanost podataka postiže se velikim brojem sigurnosnih kopija koje se mogu nalaziti na različitim mjestima. Pouzdanost ovisi o pružatelju usluga oblaka i opcijama

koje se odaberu prilikom izrade skladišta podataka, može se postići vrlo visoka razina pouzdanosti.

Uštedu u troškovima ostvaruje se korištenjem resursa po potrebi umjesto plaćanje fiksne cijene za skladište podataka. S druge strane naravno da je nekim organizacijama jeftinije izgraditi svoje skladište podataka, no za to je potrebno iskustvo i znanje.

Skladišta podataka na oblaku su dostupna u različitim oblicima, uključujući privatne, javne i hibridne oblake. Privatni oblak je namijenjen za upotrebu samo određene organizacije. Javni oblak pruža usluge i resurse dostupne svima. Hibridni oblak je kombinacija privatnog i javnog oblaka, koja omogućuje organizaciji da odabere način na koji želi da pohranjuje i pristupa svojim podacima (Awoyelu et al., 2013).

Neki od rizika i mana koji se često spominju uz skladište podataka na oblaku su sigurnost, pouzdanost, dostupnost itd. Svaka od tih stavka se može svesti na ovisnost o pružatelju usluga, ali i opcijama koje se odaberu i dogovore s pružateljem usluga tokom izgradnje skladišta podataka.

U svakom slučaju izbor između izgradnje vlastitog fizičkog skladišta podataka ili skladišta podataka na oblaku je specifičan za svaku organizaciju. Iz svega navedenog, jasno je da skladišta podataka na oblaku imaju važnu ulogu u poslovnom okruženju današnjice. Pomažu organizacijama da unaprijede svoje poslovanje i da se lakše prilagode promjenama u okruženju. Međutim, korištenje skladišta podataka na oblaku također podrazumijeva određene rizike i izazove, te je važno da organizacije odaberu pouzdane i sigurne pružatelje usluga.

6. Projektni dio: Integracija podataka pomoću Azure servisa

Microsoft Azure je platforma koja pruže različite usluge na oblaku, uključujući računarske, analitičke, skladišne i mrežne usluge. Njegova dostupnost u više regija, različite vrste računa i integracije s drugim platformama omogućuju korisnicima da izgrade i upravljaju aplikacijama na oblaku na način koji odgovara njihovim potrebama. Azure također pruža alate za razvoj aplikacija i sigurnosne mogućnosti kako bi se osigurala sigurnost aplikacija i podataka.

U ovom radu najviše će se koristiti servisi koji pomažu pri integraciji podataka te servisi koji služe za skladištenje podataka. Za to će nam biti potrebni SQL server, baza podataka, *Blob storage* i *Data Factory*.

Azure Data Factory (ADF) je servis koji se koristi za ekstrakciju, transformaciju i učitavanja podataka (engl. ETL tool). Unutar ADF-a kreiraju se cjevovodi (engl. pipeline) koji omogućuju automatizaciju cijelog ETL procesa.

Praktični primjer je rađen na zamišljenom primjeru trgovine. Trgovina sprema podatke u transakcijskoj bazi podataka. Sve veći priljev podataka onemogućava upravi da dođe do korisnih i točnih informacija. Zadatak je postojeće podatke pripremiti i integrirati u skladište podataka koristeći Azure servise na oblaku.

6.1. Izvor podataka

Podatci su dostavljeni u datoteci gdje su vrijednosti odvojene zarezom (*engl. CSV*). Jedna CSV datoteka odgovara jednoj tablici izvorišne transakcijske baze podataka.

Tablice koje se nalaze na izvoru su: *Categories, Cities, Countries, Customers, EmployeePositions, Employees, Invoice, Invoice_details, Products i Stores.*

Categories tablica predstavlja kategorije proizvoda. Definicije njenih atributa prikazane su u donjoj tablici.

Tablica 8: Tablica - Kategorije

Categories		
CategoryID	ID NOT NULL	Primarni ključ tablice.
CategoryName	Varchar(50)	Ime kategorije proizvoda.
CategoryGroup	Varchar(50)	Grupa unutar koje se nalazi kategorija proizvoda.

Cities tablica predstavlja gradove te se uglavnom koristi za određivanje lokacije kod drugih objekata.

Tablica 9: Tablica - Gradovi

Cities		
CityID	ID NOT NULL	Primarni ključ tablice.
CityName	Varchar(50)	Ime grada.
Zipcode	Varchar(50)	Poštanski broj grada.
CountryID	INT	Vanjski ključ koji referencira tablicu *Countries*. Daje informaciju u kojoj državi se nalazi određeni grad.

Tablica *Countries* sadrži popis država.

Tablica 10: Tablica - Države

Countries		
CountryID	ID NOT NULL	Primarni ključ tablice
CountryName	Varchar(50)	Ime države.
CountryCode	Varchar(50)	Kôd države.

Tablica *EmployeePositions* sadrži popis pozicija koje zaposlenici mogu imati.

Tablica 11: Tablica - Pozicije zaposlenika

EmployeePositions		
EmployeePositionID	ID NOT NULL	Primarni ključ tablice
EmployeePositionName	Varchar(50)	Ime pozicije zaposlenika.
EmployeePositionDescription	Varchar(50)	Opis pozicije zaposlenika.

Tablica *Employees* sadrži sve podatke vezane uz zaposlenika.

Tablica 12: Tablica - Zaposlenici

Employees		
EmployeeID	ID NOT NULL	Primarni ključ tablice.
FirstName	Varchar(50)	Ime zaposlenika.
MiddleInitial	Varchar(1)	Inicijal srednjeg imena zaposlenika.
Last name	Varchar(50)	Prezime zaposlenika.
BirthDate	Date	Datum rođenja zaposlenika.
Gender	Varchar(1)	Spol zaposlenika.
CityID	INT	Vanjski ključ na tablicu *Cities*, prikazuje lokaciju odakle je zaposlenik.
HireDate	datetime	Datum zaposlenja zaposlenika.
EmployeePositionID	INT	Vanjski ključ na tablicu *EmployeePositions*, prikazuje poziciju zaposlenika.

Tablica *Product* sadrži sve podatke vezane uz proizvode:

Tablica 13: Tablica - Proizvodi

Products		
ProductID	ID NOT NULL	Primarni ključ tablice.
ProductName	Varchar (50)	Ime proizvoda.
Price	Numeric (28,5)	Prodajna cijena proizvoda.
CategoryID	INT	Vanjski ključ na tablicu *Categories*, prikazuje kategoriju unutar koje se proizvod nalazi.
Class	Varchar (50)	Klasa proizvoda.
ModifyDate	datetime	Datum zadnje promjene nekog od atributa unutar tablice.
Resistant	Varchar (50)	Otpornost proizvoda na oštećenje.
IsAllergic	BIT	Može li proizvod izazvati alergijsku reakciju.
VitalityDays	INT	Broj dana koji označava rok trajanja proizvoda.
PurchasePrice	Numeric (28,5)	Nabavna cijena proizvoda.
CityID	INT	Vanjski ključ na tablicu *Cities*, prikazuje lokaciju odakle je proizvod.

Customers sadrži sve podatke vezane uz klijente.

Tablica 14: Tablica - Klijenti

Customers		
CustomerID	ID NOT NULL	Primarni ključ tablice.
FirstName	Varchar(100)	Ime klijenta.
MiddleInitial	Varchar(1)	Inicijal srednjeg imena klijenta.
LastName	Varchar(100)	Prezime klijenta.
CityID	INT	Vanjski ključ na tablicu Cities, prikazuje lokaciju odakle je klijent.
Address	Varchar(100)	Adresa na kojoj se nalazi klijent.

Invoice predstavlja informacije o izvršenoj prodaji.

Tablica 15: Tablica - Faktura

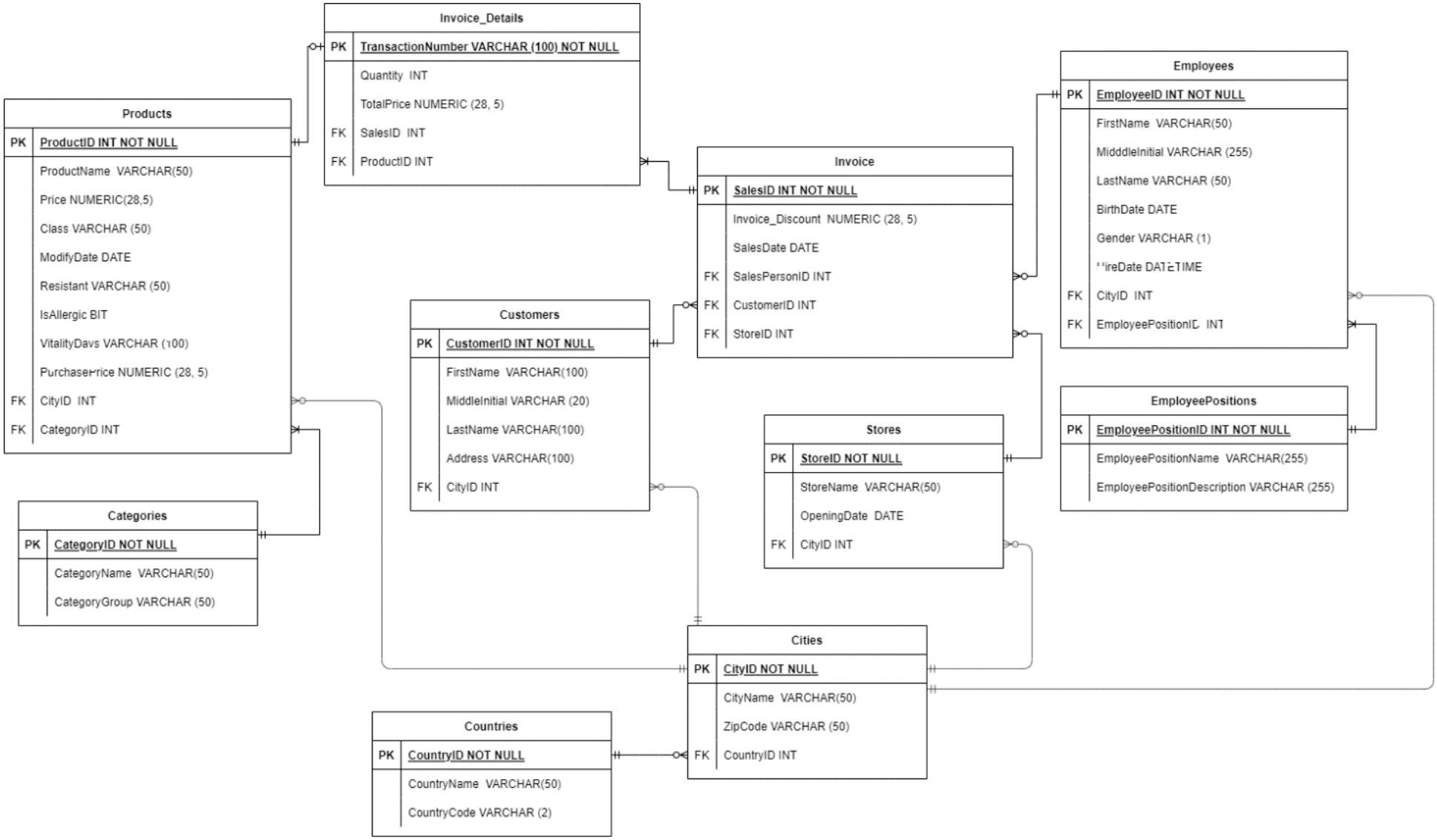
Invoice		
SalesID	ID NOT NULL	Primarni ključ tablice.
SalesPersonID	Varchar(100)	Ime zaposlenika koji je prodao proizvode.
CustomerID	Varchar(1)	Inicijal klijenta koji je kupio proizvode.
StoreID	Varchar(100)	Vanjski ključ na tablicu Stores, daje informaciju o trgovini u kojoj se desila prodaja.
SalesDate	INT	Datum prodaje.
Invoice_discount	Varchar(100)	Popust na cjelokupni račun.

Invoice Details dopunjuje tablicu Invoice na način da predstavlja izvršenu prodaju po svakoj stavki proizvoda.

Tablica 16: Tablica - Detalji fakture

Invoice Details		
TransactionNumber	ID NOT NULL	Primarni ključ tablice.
ProductID	Varchar(100)	Vanjski ključ na tablicu Products, daje informaciju o proizvodu koji se nalazi na računu.
TotalPrice	Varchar(1)	Ukupna cijena po proizvodu.
Quantity	Varchar(100)	Ukupna količina po proizvodu.
SalesID	INT	Vanjski ključ na tablicu *Invoice*, daje informaciju na kojem se računu nalazi stavka po proizvodu.

Kroz analizu podataka dolazi se do veza između tablica. Veze između tablica ostvarene su vanjskim i primarnim ključevima. Na temelju tih veza kreiran je ERA model izvorišne transakcijske baze podataka koji je prikazan na donjoj slici.



Slika 3: ERA Model - izvor

6.2. Meta podatci

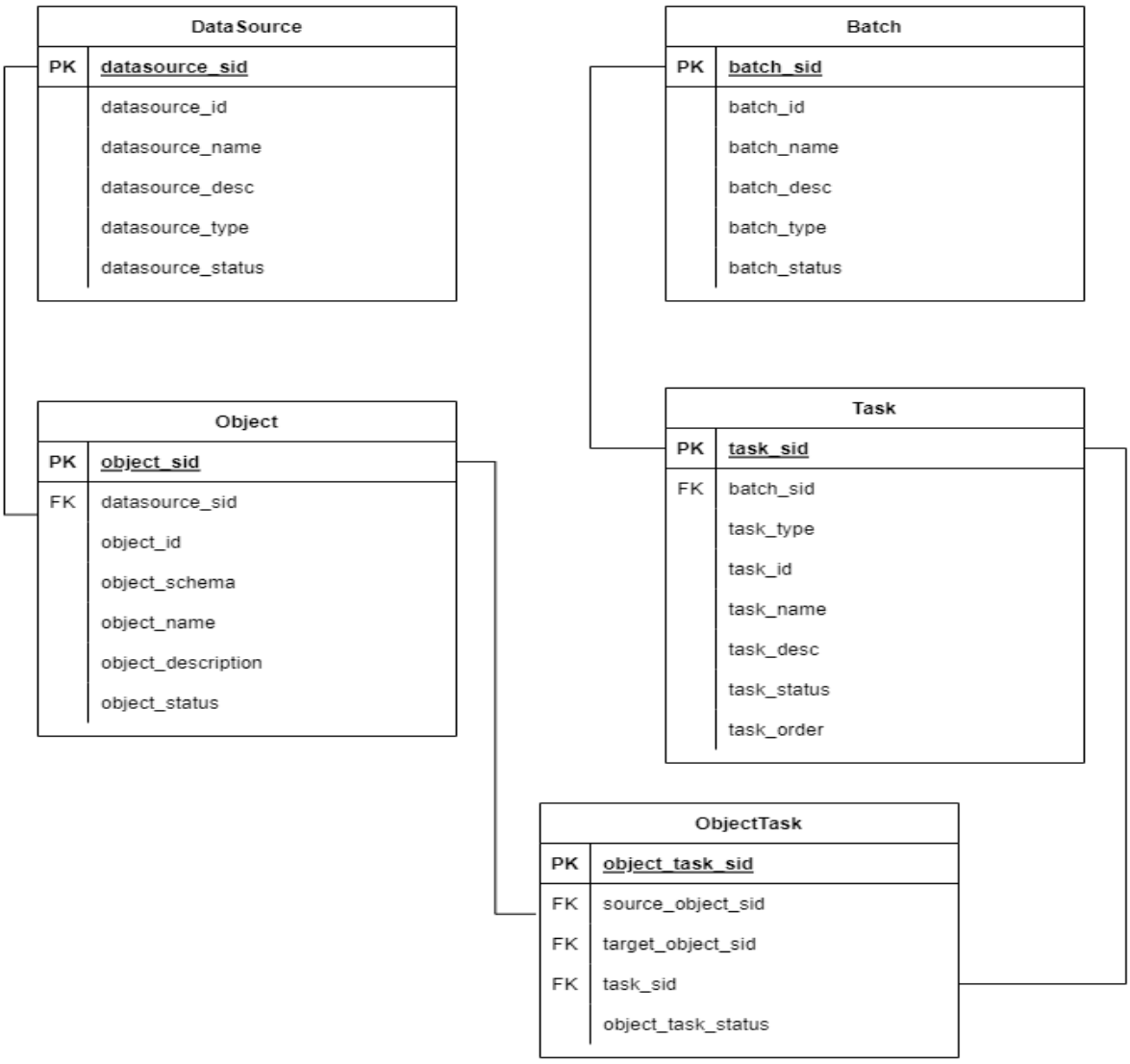
Meta podaci su podaci o podacima. Oni daju informacije o samom skladištu podataka i podacima unutar njega. Koriste ih pružatelji usluga skladišta podataka i poslovni korisnici. Prema (Huynh et al., n.d.) meta podatci se klasificiraju u sljedeće kategorije:

- Meta podatci povezani s učitavanjem i transformacijom podataka. Opisuju promjene koje su nastale nad podatcima koji dolaze s izvora te kako su podatci organizirani i spremljeni.
- Meta podatci povezani s upravljanjem podataka. Sve objekte u skladištu podataka je potrebno opisati (tablice, pogledi, indeksi itd.).
- Meta podatci koji se koriste kod postavljanja upita. Kod postavljanja upita čuvaju se podatci koji su povezani s tim upitom (npr. vrijeme odgovora). Na taj način je moguće generirati povijest upita za određenog korisnika ili grupu.

Meta podatci za ovaj projekt su organizirani u *excel* datoteci koja sadrži više stranica (engl. sheets): *DataSource, Object, Task, Batch* i *ObjectTask.*

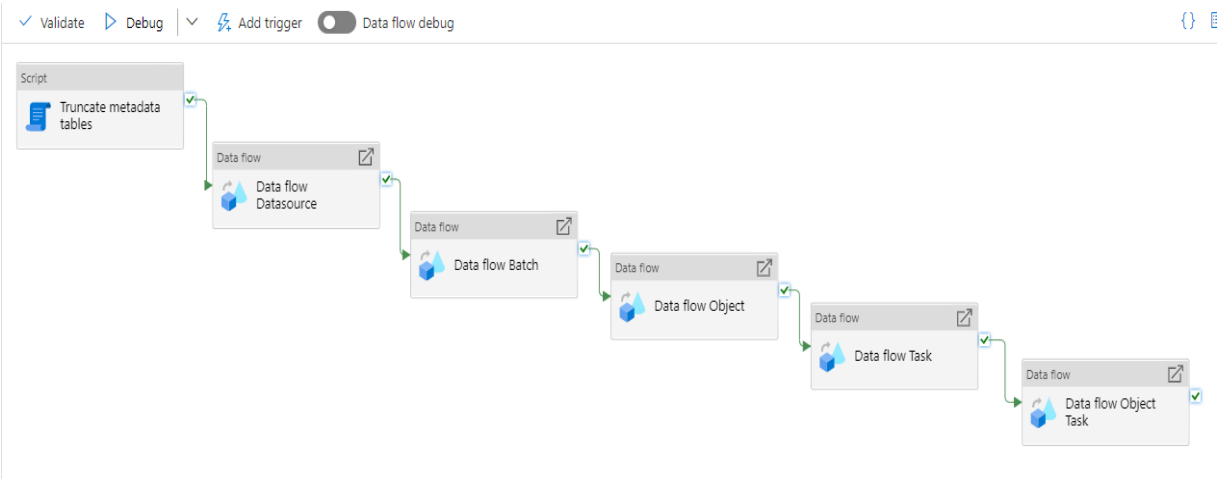
- *DataSource* – sadrži informacije o svim izvorima podataka koji se koriste u projektu, u ovom slučaju to su CSV datoteke i pripremna baza podataka.
- *Object* – sadrži sve objekte koji se koriste (tablice i datoteke).
- *Task* – sadrži sve zadatke koji se izvršavaju, poput učitavanja podataka iz CSV datoteke u tablicu te učitavanje podataka iz tih tablica u dimenzije i činjenice.
- *Batch* – logički povezuje skup zadataka iz *Task* tablice na način da ih grupira, što omogućuje jednostavno izvršavanje većeg broja zadataka po određenom redoslijedu.
- ObjectTask - povezuje zadatke iz tablice *Task* u veće logičke grupe koje se nalaze u tablici *Batch*.

Kako bi se bolje razumjeli meta podatci koji su korišteni u ovom projektu, na donjoj slici je prikazan ER model tablica koje sadrže meta podatke.



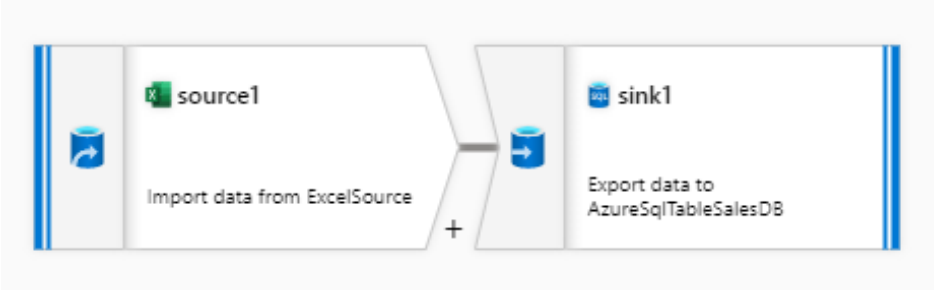
Slika 4: ER Model - meta podatci

Meta podatci se prebacuju iz *excel* datoteke u fizičke tablice na već postojećoj pripremnoj bazi podataka(engl. stage database) radi lakšeg rukovanja. Učitavanje meta podataka u bazu podataka je automatizirano kreiranjem cjevovoda unutar ADF-a.



Slika 5: Cjevovod - učitavanje meta podataka

U cjevovodu se prije učitavanja meta podataka brišu svi postojeći podatci unutar tablica koje sadrže meta podatke pomoću skripte tj. komponente *Script*. Nakon toga redom se učitavaju podatci u pojedine tablice koristeći komponentu: *Data Flow. Data Flow* komponenta služi za prebacivanje podataka iz izvora do odredišta.



Slika 6: Protok podataka - meta podatci

Data Flow za prebacivanje meta podataka iz *excel* datoteke do pripremne baze podataka je poprilično jednostavan jer ne dolazi do nikakvih transformacija. Podatci u onom obliku kakvi su na izvoru, završit će i na odredištu. Sve tablice meta podataka koriste isti *Data Flow*, to je moguće jer se imena izvorišnih datoteka i odredišnih tablica prosljeđuju dinamički kroz cjevovod.

6.3. Priprema podataka

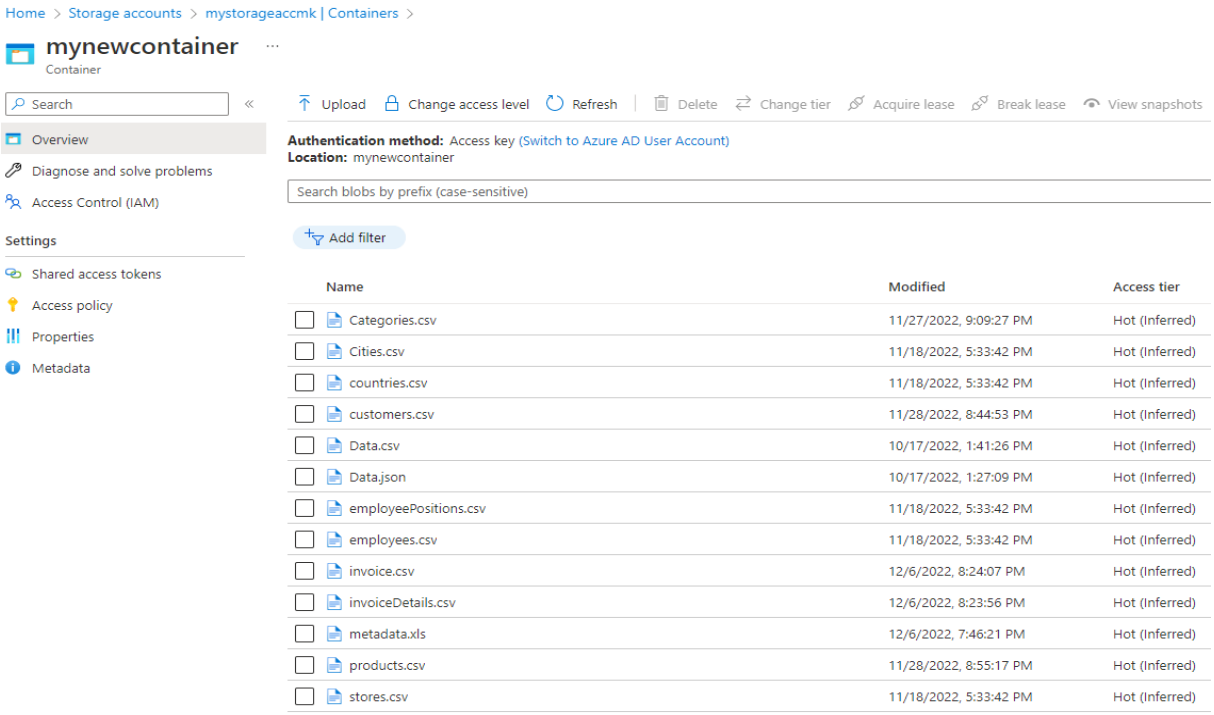
Učitavanje podataka iz izvora u pripremnu (*engl. stage*) bazu podataka se radi prije pripreme podataka. Priprema tj. transformacija je važan korak prije učitavanja podataka u skladište podataka. Poželjno se odrađuje u pripremnoj bazi podataka koristeći SQL procedure. Osim u SQL procedurama, dodatne transformacije je moguće ubaciti u cjevovod koji služi za učitavanje podataka u skladište podataka.

6.3.1. Prijenos podataka u pripremnu bazu podataka

U ovom projektu, podaci se prenose iz CSV datoteka u pripremnu bazu podataka. Prije svega, datoteke s podatcima se prebacuju na oblak. Na taj način s datotekama može raditi veći broj ljudi kojima je omogućen pristup.

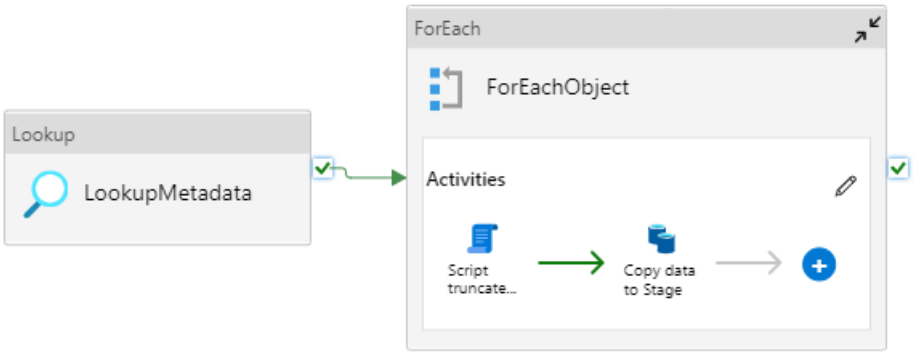
Prvo je napravljen račun za pohranu (*engl. Storage account)* u Azure portalu. Račun za pohranu se koristi za konfiguriranje i pristupanje različitim servisima skladištenja, kao i za praćenje i upravljanje troškovima skladištenja. U ovom projektu se koristi *blob* pohrana (*engl. Blob Storage*) koji služi za skladištenje velikih datoteka kao što su fotografije, videozapisi, audio snimci i drugi mediji.

Izgled kreiranog kontejnera (*engl. container*) nakon što su učitane datoteke s podacima izvora prikazan je na donjoj slici.



Slika 7: Kontejner za pohranu

Nad kontejnerom se postavljaju posebne dozvole kako bi se omogućio pristup do podataka samo osobama s odgovarajućim dopuštenjima. Nakon što se datoteke nalaze u kontejneru njima se lako pristupa i unutar drugih servisa na Azure portalu. Datotekama će se najčešće pristupati kroz ADF. U ADF-u se kreira cjevovod kako bi se podatci postavljeni na oblak učitali u pripremnu bazu podataka. Cjevovod za učitavanje podataka u pripremnu bazu podataka je prikazan je na donjoj slici.



Slika 8: Cjevovod - učitavanje podataka u pripremnu bazu

Lookup (LookupMetadata) je prva komponenta u cjevovodu koji se koristi za traženje svih potrebnih objekata za učitavanje, kao i objekata koji služe kao izvor podataka. Sve te informacije se nalaze u tablicama koje sadrže meta podatke, a donji SQL upit se koristi kako bi se došlo do tih informacija.

```
SELECT  o.object_name as srcObject,o.object_schema as srcSchema,
o2.object_name as tgtObject, o2.object_schema as tgtSchema
FROM [meta].object_task ot
JOIN [meta].task t
      ON ot.task_sid=t.task_sid
JOIN [meta].batch b
      ON t.batch_sid=b.batch_sid
JOIN [meta].object o
      ON ot.source_object_sid=o.object_sid
JOIN [meta].object o2
      ON ot.target_object_sid=o2.object_sid
WHERE 1=1
AND ot.object_task_status='ACTIVE'
AND o.object_status='ACTIVE'
AND t.task_status='ACTIVE'
AND o2.object_status='ACTIVE'
AND b.batch_status='ACTIVE'
AND b.batch_id='',pipeline().parameters.Batch_id,''
```

Sve tablice koje sadrže meta podatke se koriste za dobivanje potrebnih informacija. *WHERE* klauzula sadrži dvije vrste uvjeta. Prva vrsta filtrira sve podatke koji nisu aktivni, a druga se odnosi na atribut *Batch*_id iz tablice *Batch*. Vrijednost koja se koristi u uvjetu se dobiva dinamički putem parametra u ADF-u, što znači da se dobivaju rezultati za točno određenu grupu zadataka (engl. batch). Rezultat SQL upita za *batch_id = 'STAGE_LOAD'* prikazan je na donjoj slici.

Results Messages

	srcObject	srcSchema	tgtObject	tgtSchema
1	Categories	N/A	Categories	STAGE_IN
2	Cities	N/A	Cities	STAGE_IN
3	countries	N/A	Countries	STAGE_IN
4	customers	N/A	Customers	STAGE_IN
5	employeePositions	N/A	EmployeePositions	STAGE_IN
6	employees	N/A	Employees	STAGE_IN
7	products	N/A	Products	STAGE_IN
8	stores	N/A	Stores	STAGE_IN
9	invoice	N/A	Invoice	STAGE_IN
10	invoiceDetails	N/A	Invoice_Details	STAGE_IN

Slika 9: Grupa zadataka - STAGE LOAD

STAGE_LOAD je grupa zadataka koje je potrebno odraditi kod učitavanja podataka iz CSV datoteka u pripremnu bazu podataka.

Sljedeća komponenta u cjevovodu je *ForEach (ForEachObject)* petlja koja za svaki rezultat iz *Lookup-a* izvršava sljedeće aktivnosti:

- Skripta koja čisti tablicu u kojoj će biti smješteni izvorni podaci prije nego što se oni učitaju (tzv. truncate)
- Komponenta za kopiranje podataka (*engl. Copy Data*) koja dinamički određuje imena izvornih i odredišnih objekata te kopira podatke iz izvornih objekata navedenih u meta podacima u odredišne objekte navedene u meta podacima.

6.3.2. Transformacija podataka unutar pripremne baze podataka

Analizirajući podatke, otkrivena je pogreška u tablici *Countries*. Atribut *CountryName* koji predstavlja ime države i atribut *CountryCode* koji predstavlja kôd države se ne podudaraju. Preuzeti su podaci o imenima država i njihovim kôdovima s interneta. Na temelju tih podataka,

stvorena je nova tablica *country_codes*. Koristeći tu tablicu, SQL naredbom ažurirana je tablica *Countries* s ispravnim vrijednostima kôdova države.

```
UPDATE c
SET c.[CountryCode]=cd.[CountryCode]
FROM [STAGE_IN].[Countries] c
  JOIN dbo.country_codes cd
      ON c.CountryName=cd.CountryName
```

Tablica Products sadrži atribute *IsAllergic*, *Resistant* i *VitalityDays*, gdje postoje vrijednosti zapisane kao tekstualni niz 'NULL'. Te vrijednosti se pretvaraju u NULL, koji predstavlja polje bez vrijednosti.

```
  UPDATE [STAGE_IN].[Products]
  SET IsAllergic=NULL
  WHERE IsAllergic='NULL'

  UPDATE [STAGE_IN].[Products]
  SET  Resistant=NULL
  WHERE  Resistant='NULL'

  UPDATE [STAGE_IN].[Products]
  SET  VitalityDays=NULL
  WHERE VitalityDays='NULL'
```

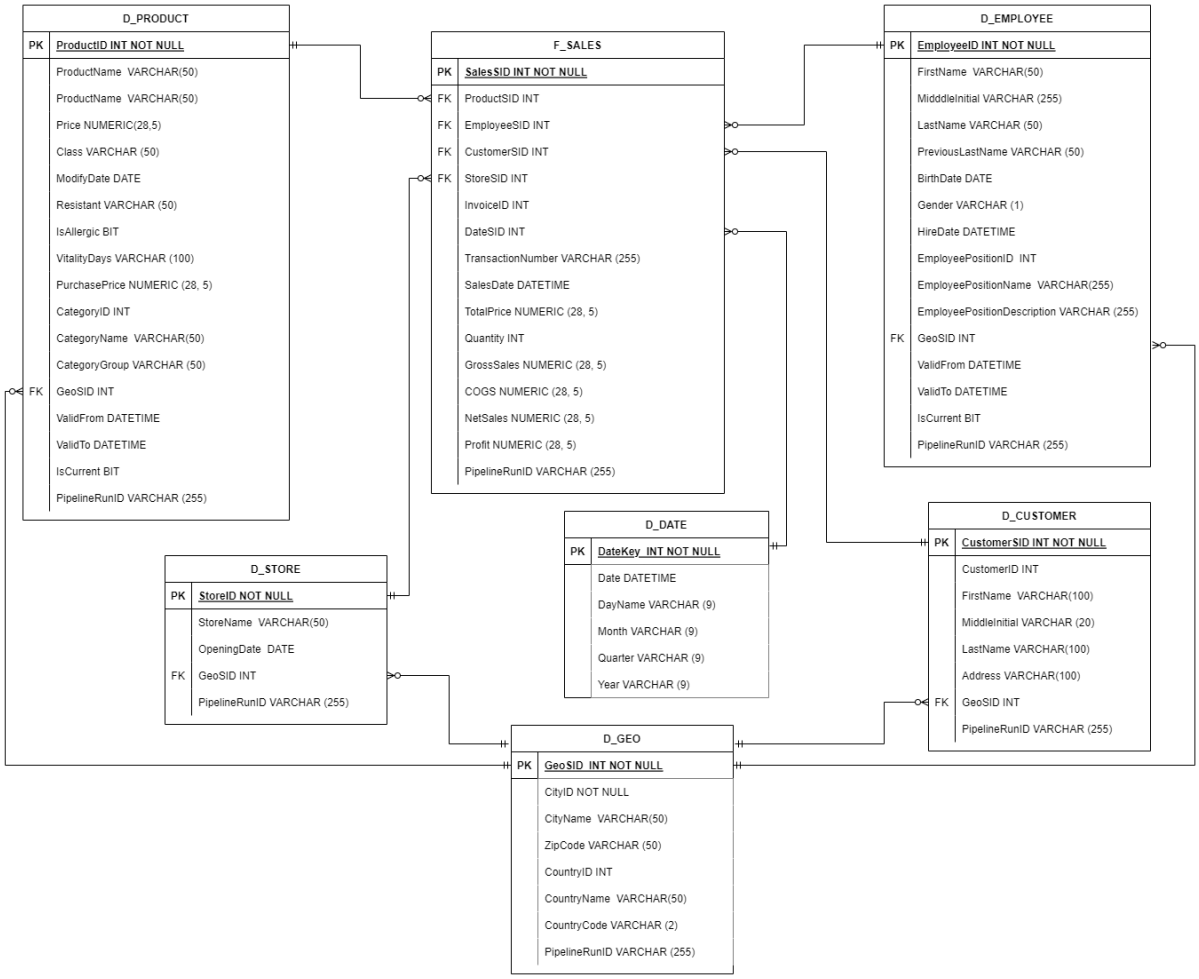
Tablica Products sadrži atribut Price gdje se u nekim slučajevima nalazi vrijednost manja od nule. Budući da cijena proizvoda ne može biti negativna, ona se zamjenjuje vrijednošću atributa *PurchasePrice*. Na taj način, proizvod s pogrešno upisanom cijenom neće previše utjecati na završne izračune jer će mu prodajna cijena biti jednaka kupovnoj cijeni.

```
  UPDATE [STAGE_IN].[Products]
  SET Price=PurchasePrice
  WHERE Price <=0
```

6.4. Učitavanje podataka u skladište podataka

Skladište podataka usmjereno je na praćenje prodaje proizvoda i sastoji se od šest dimenzija: *D_PRODUCT, D_STORE, D_EMPLOYEE, D_CUSTOMER, D_GEO, D_DATE*. U

skladištu podataka također se nalazi i jedna činjenična tablica: F_SALES. ERA model skladišta podataka prikazan je na donjoj slici.

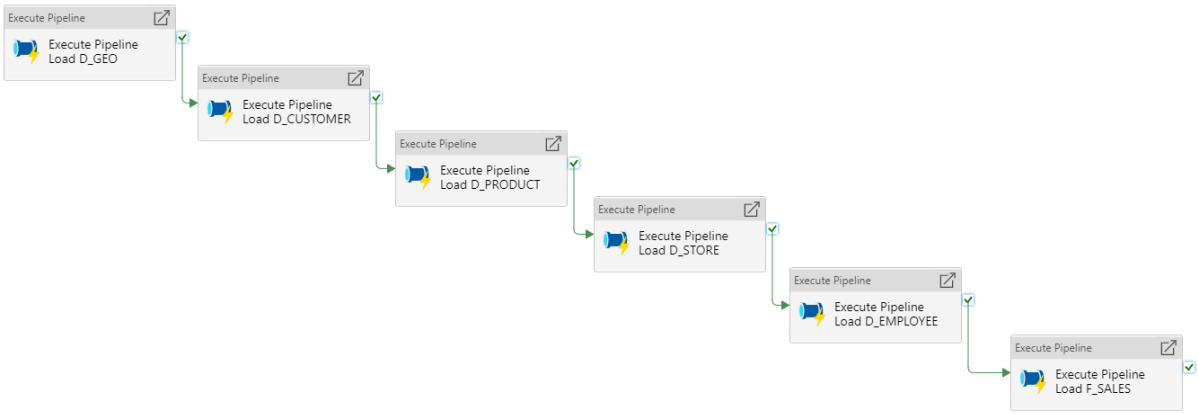


Slika 10: Shema skladišta podataka

Model koji je kreiran baziran je na modelu zvijezde, s iznimkom dimenzije *D_GEO*. *D_GEO* dimenzija se koristi za utvrđivanje lokacije kod ostalih dimenzija pomoću vanjskog ključa, dok se dimenzije *D_PRODUCT, D_STORE, D_EMPLOYEE, D_CUSTOMER, D_DATE* koriste kao vanjski ključevi u činjeničnoj tablici *F_SALES*. Ovakav model omogućuje dobivanje detaljnih informacija o pojedinačnoj prodaji proizvoda, kao što su naziv proizvoda, količina prodanog proizvoda, cijena proizvoda, datum prodaje, ime i prezime osobe koja je izvršila prodaju (*D_EMPLOYEE*), ime i prezime osobe koja je kupila određeni proizvod (*D_CUSTOMER*) te naziv i lokaciju trgovine u kojoj se odvila prodaja (*D_STORE*).

Za učitavanje dimenzijskih tablica i činjenične tablice napravljen je glavni cjevovod koji po određenom redoslijedu pokreće učitavanje pojedinačnih tablica. Na taj način svi cjevovodi

se mogu pokrenuti jednim klikom, što smanjuje vjerojatnost pogreške i ubrzava proces učitavanja podataka. Dimenzija *D_DATE* je preuzeta s interneta te se neće mijenjati pa se i ne nalazi u glavnom cjevovodu. Arhitektura glavnog cjevovoda prikazana je na donjoj slici.



Slika 11: Cjevovod - učitavanje skladišta podataka

Komponenta koja se koristi na gornjoj slici je *Execute Pipeline*, ona ima zadatak pokretanja odvojenih cjevovoda. Prvo se pokreće cjevovod za učitavanje dimenzije *D_GEO*, jer se ona koristi za određivanje lokacije kod ostalih dimenzija pomoću vanjskog ključa. Nakon toga se pokreću cjevovodi za učitavanje ostalih dimenzija (*D_PRODUCT, D_STORE, D_EMPLOYEE, D_CUSTOMER),* redoslijed nije bitan. Na kraju se pokreće cjevovod za učitavanje činjenične tablice *F_SALES*. Učitavanje činjenične tablice *F_SALES* se vrši na kraju jer je nužno prethodno učitati sve dimenzije kako bi se mogle referencirati u činjeničnoj tablici koristeći surogat ključeve.

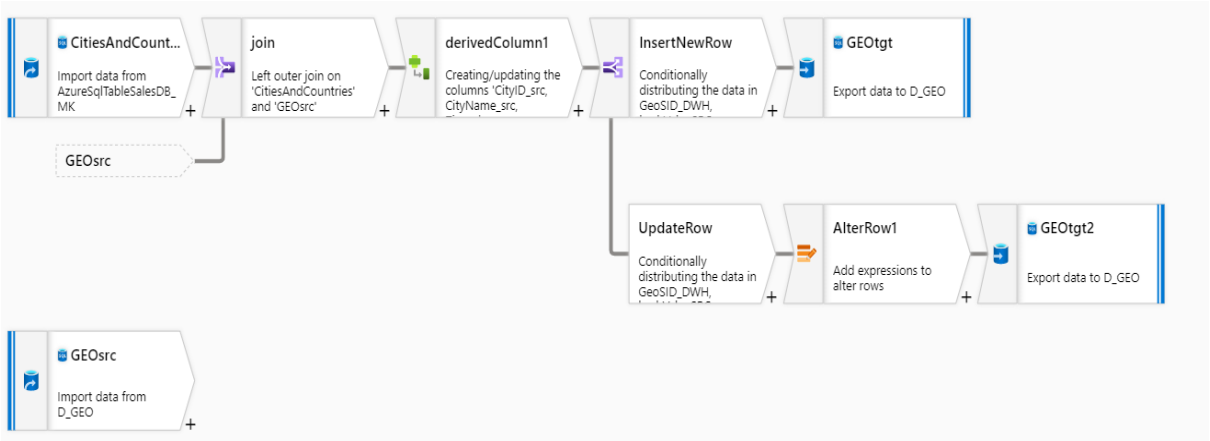
U nastavku su detaljnije opisani svi cjevovodi koji služe za učitavanje podataka u pojedine dimenzije i činjeničnu tablicu.

6.4.1. Dimenzija D_GEO

Učitavanje podataka u dimenziju *D_GEO* je automatizirano kreiranjem cjevovoda u ADF-u. Ovaj cjevovod služi za automatsko preuzimanje podataka iz izvora podataka te ih učitava u dimenziju *D_GEO*. *D_GEO* ne sprema povijest podataka, što znači da se ne spremaju sve promjene koje se događaju na podacima. Međutim, ako se u dimenziju učitava podatak koji već postoji u skladištu podataka, on se automatski ažurira. Provjera postoji li podatak u skladištu podataka se vrši preko prirodnog ključa, što omogućuje da se izbjegne

dupliciranje podataka. Opisana situacija je poznata kao SCD 1 (sporo mijenjajuća dimenzija 1) i koristi se za održavanje ažurnosti podataka u dimenzijama.

Arhitektura učitavanja dimenzije D_GEO prikazana je na donjoj slici, što omogućuje bolje razumijevanje procesa učitavanja podataka.



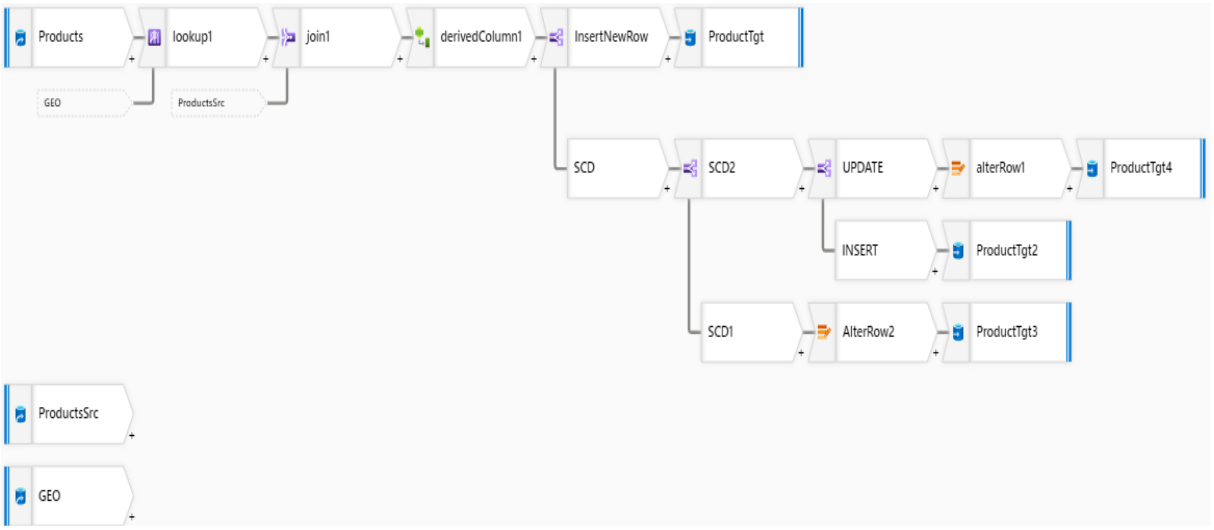
Slika 12: Cjevovod - učitavanje D_GEO

Prva komponenta u cijevi predstavlja izvor podataka, a to su tablice iz izvora: *Cities* i *Countries*. Kako bi se osiguralo upravljanje s podatcima koji se već nalaze u skladištu podataka, koristi se komponenta *join* pomoću koje se spajaju podatci iz skladišta podataka i izvora podataka preko prirodnog ključa *CityID*. U komponenti *derived column*, izračunavaju se dvije hash vrijednosti, jedna na temelju podataka iz izvora i jedna na temelju podataka iz odredišta (dimenzije), za jedan slog podataka. Te hash vrijednosti se koriste u komponenti *conditional split* (InsertNewRow / UpdateRow) kako bi se utvrdilo postoji li razlika između podataka na izvoru i odredištu. Ukoliko se hash vrijednosti razlikuju, što znači da su i neki atributi različiti, podatci u skladištu podataka se ažuriraju. Ako podatci ne postoje na odredištu, oni se automatski ubacuju u skladište podataka. Ukoliko podatak postoji na odredištu, no hash vrijednosti izvorišnih i odredišnih podataka su jednake, što znači da su i svi atributi identični, tada se ne radi ništa tj. podatak se preskače jer se ne zadovoljava ni jedan uvjet u komponenti *conditional split*.

6.4.2.Dimenzija D_PRODUCT

Učitavanje podataka u dimenziju D_PRODUCT je automatizirano putem kreiranja cjevovoda u ADF-u. Ovaj cjevovod prati povijest atributa: *Price*, *ProductName* i *PurchasePrice*, koristeći metodu SCD 2. Ostali atributi se samo ažuriraju u skladištu ukoliko se promijene na

izvoru podataka (SCD 1). Arhitektura cjevovoda koja se koristi za učitavanje podataka u dimenziju *D_PRODUCT* je prikazana na donjoj slici.

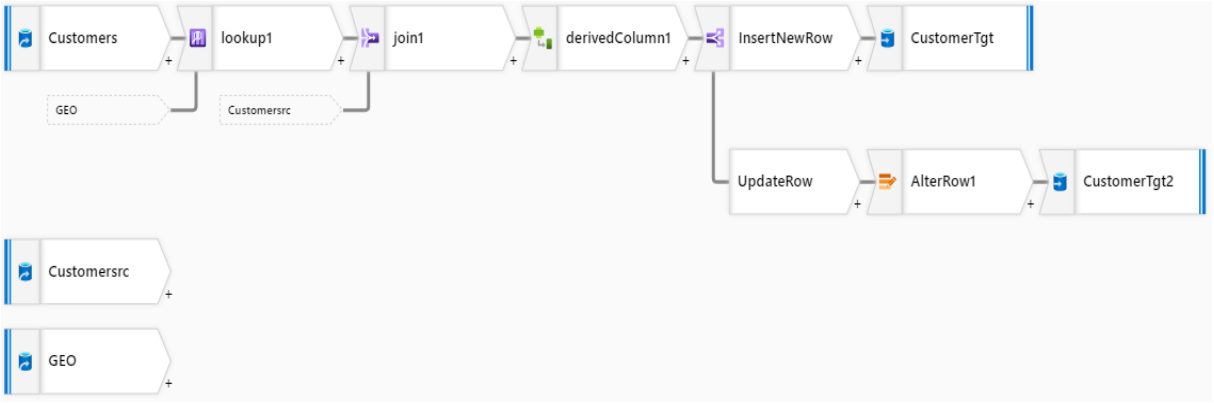


Slika 13: Cjevovod - učitavanje D_PRODUCT

U ovom cjevovodu, prva komponenta koristi tablicu Products kao izvor podataka. *Lookup* komponenta pretražuje surogat ključ u dimenzijskoj tablici *D_GEO* pomoću *CityID* ključa koji je prirodni ključ u *D_GEO* tablici. *Join* komponenta spaja podatke iz izvora s podatcima koji se već nalaze u skladištu u dimenziji *D_PRODUCTS*. Komponenta *derived column* koristi se za izračunavanje dvije hash vrijednosti atributa s odredišta i izvora. Prve hash vrijednosti odredišta i izvora će se dobiti na temelju atributa kojima se prati povijest (SCD 2), a druge hash vrijednosti se dobivaju na temelju atributa kojima se ne prati povijest (SCD 1). Komponenta *conditional split* (InsertNewRow / SCD) se koristi za dijeljenje podataka s obzirom na to postoje li na izvoru. Ukoliko podatak ne postoji na izvoru, podatak se ubacuje u skladište podataka. Ukoliko podatak postoji na izvoru, podatci ponovno ulaze u *conditional split* (SCD 2 / SCD 1) te se dijele s obzirom na to radi li se o SCD 1 ili SCD 2. SCD 1 podrazumijeva ažuriranje podataka na odredištu, dok SCD2 podrazumijeva ažuriranje podataka na odredištu s novim vrijednostima u poljima *ValidTo* i *IsCurrent*, podatak više neće biti aktivan. Osim ažuriranja, dodaje se novi redak s trenutno aktualnim vrijednostima atributa, te će taj podatak biti aktivan.

6.4.3.Dimenzija D_CUSTOMER

Dimenzija *D_CUSTOMER* koristi tip 1 sporomijenjajuće dimenzije (SCD 1) u slučaju promjena nad podatcima na izvoru. Arhitektura cjevovoda koji služi za učitavanje podataka u dimenziju *D_CUSTOMER* prikazana je na donjoj slici.

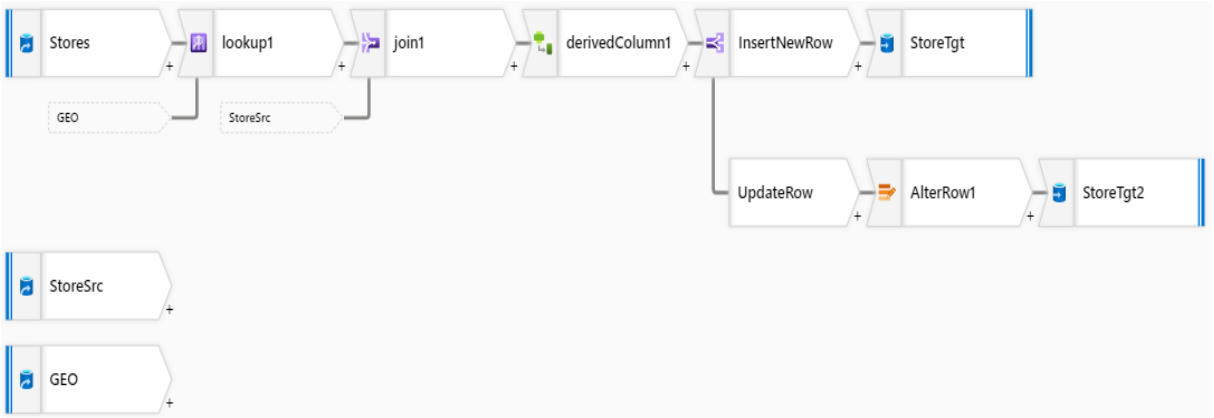


Slika 14: Cjevovod - učitavanje D_CUSTOMER

Prva komponenta prikazuje izvor podataka, a to je tablica *Customers*. *Lookup* komponenta se koristi za pretraživanje odgovarajućeg surogat ključa iz *D_GEO* tablice koristeći prirodni ključ. *Join* komponenta se koristi za spajanje podataka iz izvora s podacima koji se već nalaze na odredištu u dimenziji *D_CUSTOMER*. Spajanje se provodi preko prirodnog ključa. Komponenta *derived column* se koristi za izračunavanje hash vrijednosti za svaki podatak na temelju vrijednosti atributa s izvora i na temelju vrijednosti atributa s odredišta. *Conditional split* (InsertNewRow / UpdateRow) komponenta ima dva uvjeta. Ukoliko podatak ne postoji na odredištu, on se ubacuje u dimenziju. Ukoliko podatak postoji na odredištu uspoređuju se hash vrijednosti i ako su različite podatci se ažuriraju, a ukoliko su hash vrijednosti jednake ne događa se ništa tj. podatak se preskače jer se radi o duplikatu.

6.4.4. Dimenzija D_STORE

Proces učitavanja podataka u dimenziju D_STORE ne uključuje praćenje povijesti promjena atributa. Ukoliko se promijeni bilo koji atribut, on se samo ažurira u skladištu podataka (SCD 1). Arhitektura cjevovoda koji se koristi za učitavanje dimenzije D_STORE je prikazana na donjoj slici.

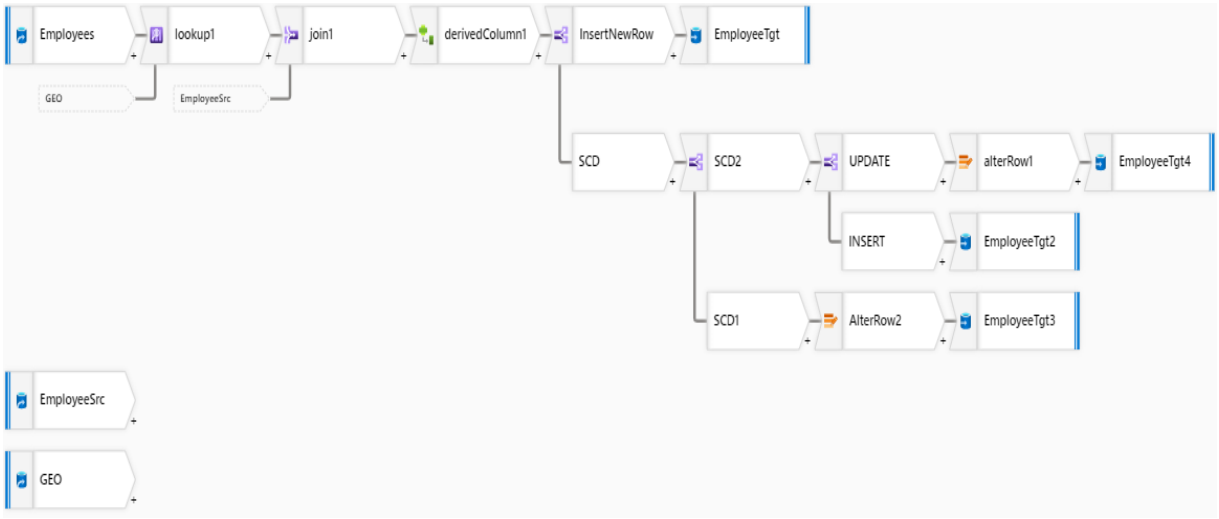


Slika 15: Cjevovod - učitavanje D_STORE

Prva komponenta predstavlja izvor podataka, a to je tablica s izvora *Stores*. *Lookup* komponenta se koristi za pretraživanje odgovarajućeg surogat ključa iz dimenzije *D_GEO* koristeći prirodni ključ. Podatci iz izvora se spajaju s podatcima koji se već nalaze u dimenziji *D_STORE* koristeći prirodni ključ pomoću komponente *join*. Komponenta *derived column* se koristi za izračunavanje hash vrijednosti podataka na temelju vrijednosti atributa s izvora i na temelju vrijednosti atributa s odredišta. *Conditional split* (InsertNewRow / UpdateRow) komponenta ima dva uvjeta. Ukoliko podatak ne postoji na odredištu, on se ubacuje u dimenziju *D_STORE*. Ukoliko podatak postoji na odredištu te se hash vrijednosti razlikuju, podatak se ažurira unutar dimenzije *D_STORE*. Ukoliko podatak postoji na odredištu, ali se hash vrijednosti ne razlikuju, podatak se preskače i ne radi se ništa.

6.4.5. Dimenzija D_EMPLOYEE

Dimenzija *D_EMPLOYEE* čuva povijest za atribute *EmployeePositionName*, *EmployeePositionID*, *GeoSID*. Za praćenje tih atributa koristi se tip 2 sporomijenjajuće dimenzije (SCD2). Atribut *LastName* koristi SCD 6 za praćenje povijesti. Svi ostali atributi su tipa SCD 1, odnosno ne čuvaju povijesne podatke. Arhitektura cjevovoda koji služi za učitavanje podataka u dimenziju *D_EMPLOYEE* prikazana je na donjoj slici.



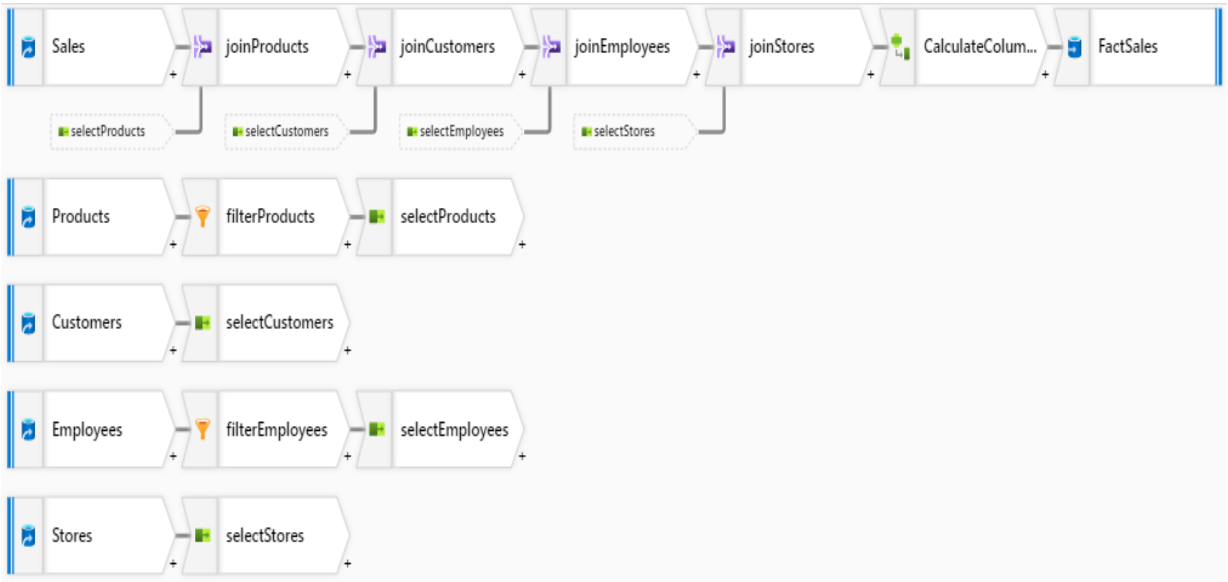
Slika 16: Cjevovod - učitavanje D_EMPLOYEE

Izvor podataka za dimenziju *D_EMPLOYEE* je izvorišna tablica *Employees*. *Lookup* se koristi za pronalazak odgovarajućeg ključa unutar dimenzije *D_GEO* na temelju prirodnog ključa *CityID*. *Join* spaja podatke iz izvora i već postojeće podatke unutar dimenzije *D_EMPLOYEE* na temelju prirodnog ključa. Unutar komponente *derived column* izračunavaju se dvije hash vrijednosti na temelju vrijednosti atributa s izvora i vrijednosti atributa s odredišta. *Conditional split* (InsertNewRow / UpdateRow) komponenta ima dva uvjeta. Kod prvog uvjeta podatci se ubacuju na odredište u slučaju kada podatak ne postoji na odredištu. U drugom slučaju podaci se preusmjeravaju na SCD2 ili SCD1, ovisno o atributima koji su drugačiji na izvoru tj. njihovim hash vrijednostima.

Ukoliko su hash vrijednosti drugačije nad atributima kojima je potrebno pratiti povijest, stari postojeći podatak se ažurira. Polja koja se ažuriraju su *ValidTo* i *IsCurrent*, podatak više neće biti aktivan. Nakon ažuriranja dodaje se novi aktivan podatak. Ukoliko se promjena dogodila na atributu *LastName*, prošla vrijednost se dodatno zapisuje u atribut *PreviousLastName*. Ukoliko su hash vrijednosti atributa drugačije nad onima kojima nije potrebno pratiti povijest, radi se obično ažuriranje podataka na odredištu SCD1.

6.4.6.Činjenica F_SALES

Nakon učitavanja svih dimenzijskih tablica dolazi na red činjenična tablica *F_SALES*. Činjenična tablica će biti fokus kod svih izračuna iz kojih se dolazi do bitnih informacija. Arhitektura cjevovoda koji služi za učitavanje podataka u činjeničnu tablicu *F_SALES* prikazana je na donjoj slici.



Slika 17: Cjevovod - učitavanje F_SALES

Izvor podataka činjenice *F_SALES* su tablice *Invoice* i *Invoice_details*. Osim podataka s izvora u činjeničnoj tablici se nalaze i vanjski ključeve na dimenzije. Tako da se za izvor koriste i sve već napunjene dimenzije. Dimenzije koje koriste tip 2 sporomijenjajuće dimenzije se filtriraju na način da se uzmu samo trenutno aktivni podatci (*IsValid=TRUE*). Sve dimenzije nakon komponente izvora, koriste komponentu *select* kako bi se odabrali samo atributi koji su potrebni, a to su prirodni ključ i surogat ključ iz dimenzije. Dodatno iz tablice D_PRODUCT uzimaju se i vrijednosti atributa *Price* i *PurcashePrice* jer se koriste u komponenti *derived column* (*CalculatedColumns*).

Derived column (*CalculatedColumns*) komponenta služi za izračunavanje vrijednosti atributa prije nego što oni mogu biti ubačeni u činjeničnu tablicu. Atributi koji se izračunavaju su:

- *GrossSales*: $Quantity * Price$
  - Predstavlja ukupnu prodaju po prodanom proizvodu. Izračunava se umnoškom količine proizvoda i cijene proizvoda.
- *PurchasePriceTotal*: $Quantity * PurchasePrice$
  - Predstavlja ukupnu kupovnu cijenu prodanog proizvod. Izračunava se umnoškom količine proizvoda i kupove cijene proizvoda.
- *GrossSalesWithDiscount*: $GrossSales * InvoiceDiscount$
  - Predstavlja ukupnu prodaju po prodanom proizvodu, umanjenu za popust. Izračunava se umnoškom ukupne prodaje po proizvodu i decimalno zapisanog popusta.

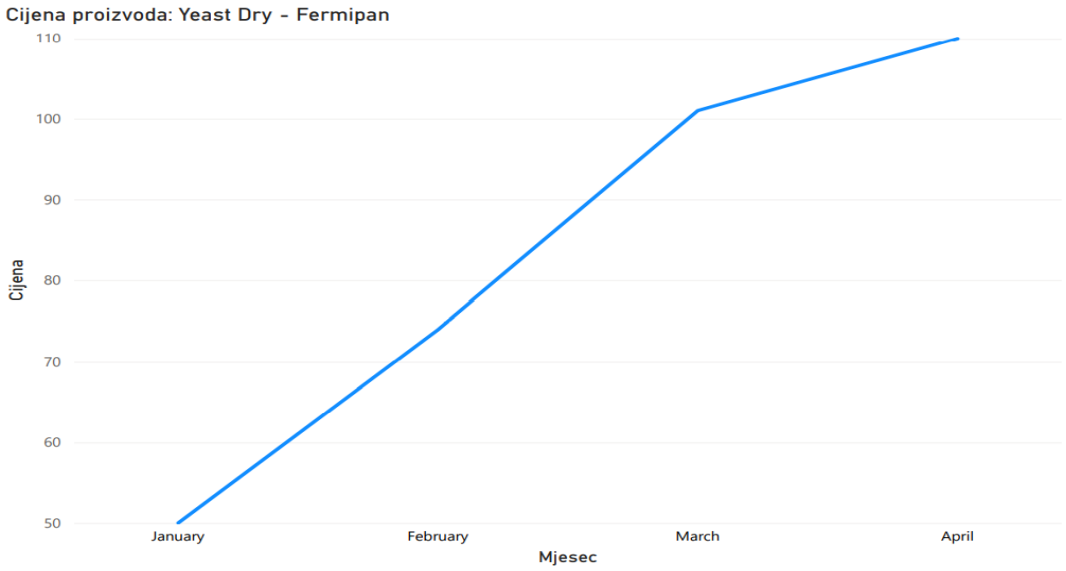
- *Profit*: $GrossSalesWithDiscount - PurchasePriceTotal$
    - Predstavlja ukupnu zaradu po proizvodu. Izračunava se kao ukupna prodaja po proizvodu s uračunatim popustom umanjena za ukupnu kupovnu cijenu proizvoda.

Atributi koji se navode kao potrebni za izračun prije učitavanja u činjeničnu tablicu su važni za stvaranje izvještaja. Izračuni se moraju napraviti prije nego što se podaci učitaju u skladište podataka, jer bi u suprotnom izvještavanje bilo sporije ako bi se izračuni radili po potrebi tijekom kreiranja izvještaja.

7. Vizualizacija podataka

Podatci u skladištu podataka se najčešće prikazuju uz neki od alata za vizualizaciju podataka. Prikaz podataka u grafičkom ili slikovnom formatu omogućuje lakše razumijevanje i interpretaciju podataka. U ovom projektu koristi se *PowerBI* kako bi se dočaralo na koji način se mogu koristiti podatci jednom kada se oni nađu u skladištu podataka. Kako je projektni dio rada usmjeren k proizvodima tako će i grafovi prikazani u nastavku prvenstveno prikazivati informacije o proizvodima.

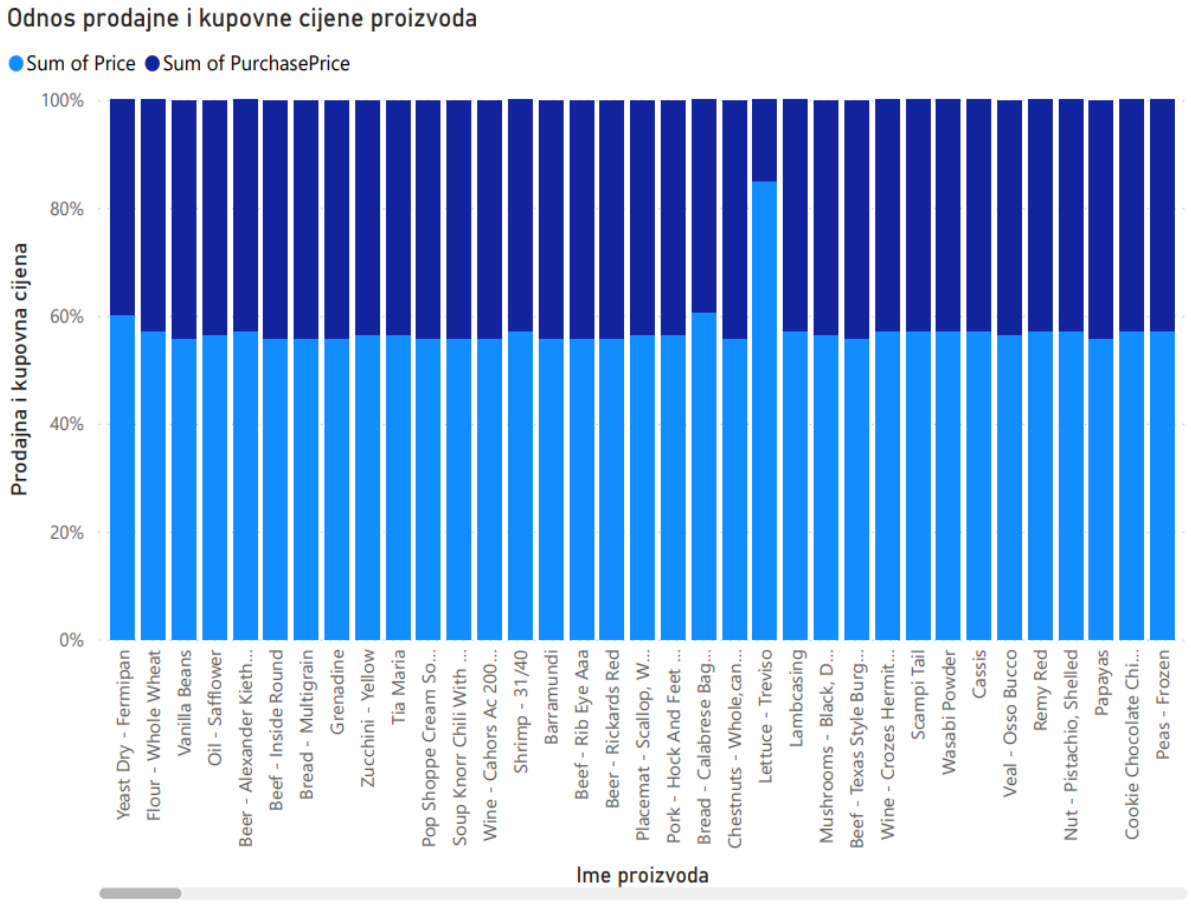
Prvi graf prikazuje cijenu proizvoda *Yeast Dry – Fermipan* po mjesecima, cijena je jedan od atributa čija se povijest čuva (SCD 2).



Slika 18: Graf : cijena proizvoda

Iz grafa se zaključuje kako se cijena proizvoda mijenjala u siječnju, veljači, ožujku i travnju. Vrlo lako se primjećuje kako je proizvod poskupljivao iz mjeseca u mjesec.

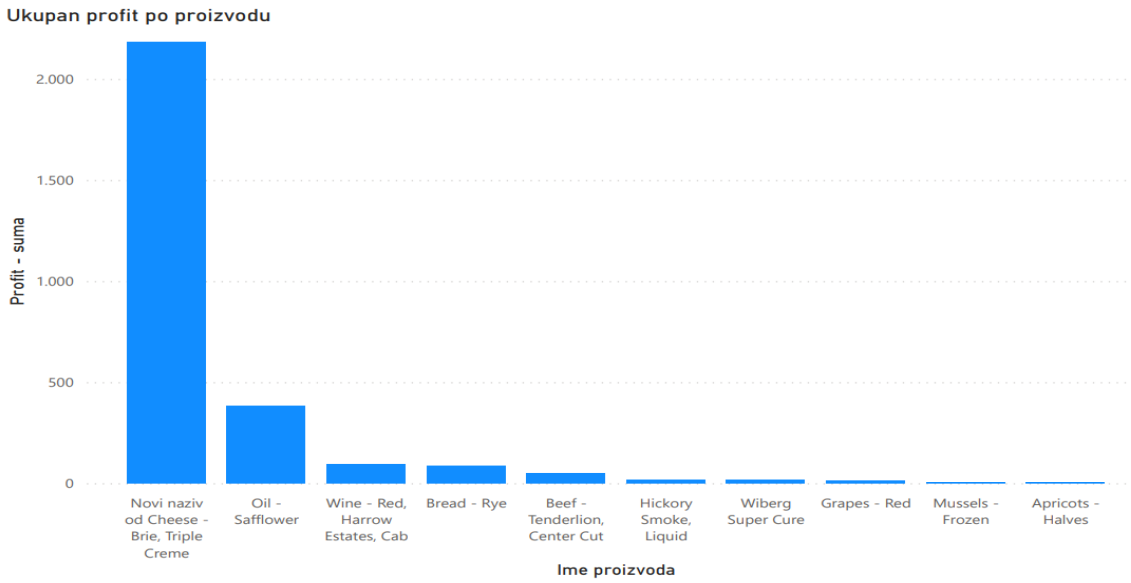
Donji graf prikazuje odnos kupovne i prodaje cijene proizvoda. Proizvodi su sortirani na način da su prvo prikazani oni s najvećom kupovnom cijenom



Slika 19: Graf - odnos kupovne i prodajne cijene

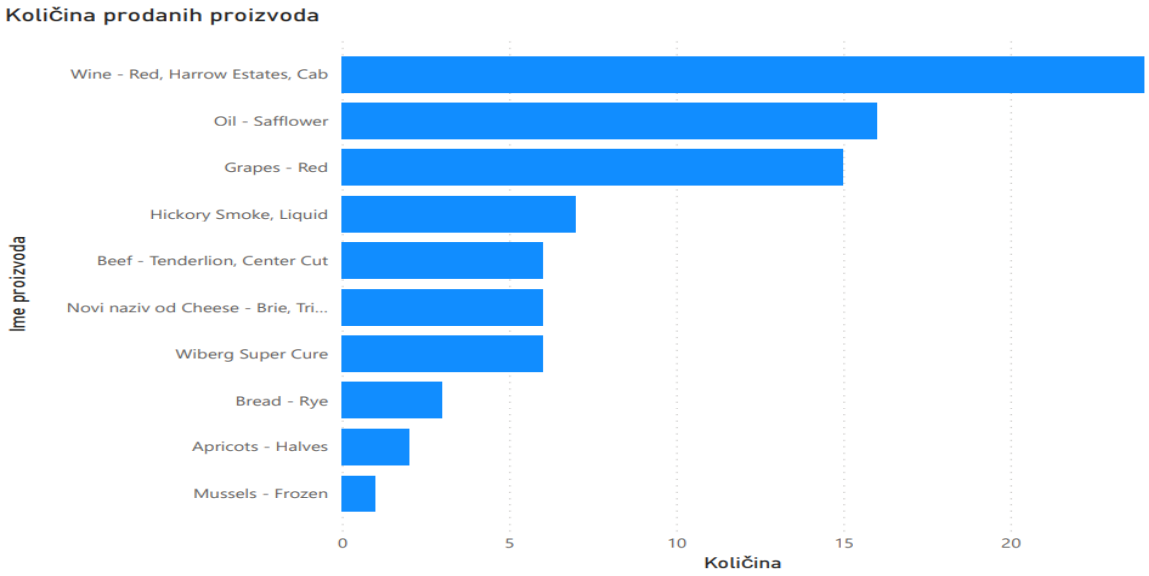
Iz gornjeg grafa se može doći do proizvoda na kojima se ostvaruje najmanji ili najveći profit. Svi proizvodi imaju veću prodajnu cijenu od kupovne što je i očekivano.

Sljedeći prikazan graf na slici 20 prikazuje ukupan profit po proizvodu. Proizvodi su sortirani od onog koji je donio ukupno najveću zaradu prema onima s manjom ukupnom zaradom.



Slika 20: Graf - ukupan profit po proizvodu

Na donjoj slici prikazan je graf koji prikazuje ukupnu količinu prodanih proizvoda.

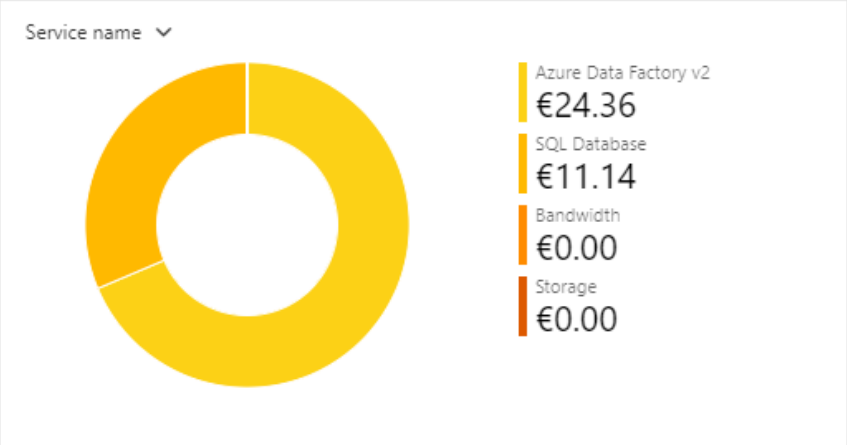


Slika 21: Graf - količina prodanih proizvoda

Svi prikazani grafovi daju važne informacije za tvrtku koja se bavi prodajom. Moguće je doći do još mnogo tipa grafova s drugačijim informacijama, ovdje su prikazani oni koji su lako razumljivi i intuitivni kako bi se prikazale mogućnosti vizualizacije podataka jednom kada se oni nalaze u skladištu podataka.

8. Analiza potrošnje

Svakako valja naglasiti još jednom da je izbor servisa kod integracije podataka specifičan za svaku organizacija. U ovom poglavlju će biti prikazani nastali troškovi tokom rada na cjelokupnom projektu. Na donjoj slici grafički je prikazan iznos troškova po servisima.



Slika 22: Sveukupni troškovi

Na servis ADF potrošeno je 24,36 eura, a na SQL bazu podataka 11,14 eura. Cijene su vrlo male, ali tokom izrade projekta nije korištena velika količina podataka baš iz razloga kako bi troškovi bili što manji.

Za izradu projekta korištena je samo jedna baza podataka radi uštede. Trošak baze podataka iznosi 11,14 eura. Odabrana je osnovna opcija s minimalnom procesnom snagom , te maksimalnom pohranom 2GB.

Detaljan prikaz troškova na ADF-u prikazan je na donjoj slici.

Service name	Meter	Cost
Azure Data Factory v2	vCore	€21.88
Azure Data Factory v2	Cloud Data Movement	€2.21
Azure Data Factory v2	Cloud Orchestration Activity Run	€0.26
Azure Data Factory v2	Cloud Pipeline Activity	<€0.01
Azure Data Factory v2	Cloud Monitoring Operations	€0
Azure Data Factory v2	Cloud Read Write Operations	€0

Slika 23: ADF troškovi

vCore predstavlja resurse koji se koriste kod procesuiranja podataka u podatkovnim tokovima, a naplaćuje se 0,274$ po satu za jedan *vCore*. Trošak *vCore-a* za izradu projekta je bio 21,88 eura.

Cloud Data Movement označuje prebacivanje podataka s jednog mjesta na drugo, te se naplaćuje 0,25$ po satu ukoliko se koristi jedan DIU (engl. data integration unit) koji predstavlja jednu jedinicu za podatkovnu integracija tj. spoj procesora, memorije i mrežnih resursa. Na ovom projektu za ovaj resurs je potrošeno 2,21 euro.

Cloud Orchestration Activity Run se naplačuje 1$ po 1000 pokretanja. Tu spadaju pokretanja i debugiranje svih aktivnosti i okidača koji se nalaze u procesu tj. cjevovodu. Plaćeno je 0,26 eura za ove resurse. Što zvuči poprilično malo, ali s obzirom da su se procesi pokretali mali broj puta te procesi na sadrže velik broj aktivnosti, cijena je očekivana.

Cloud Pipeline Activity Run predstavlja vrijeme koje je potrošeno za izvršavanje cjevovoda. Naplaćuje se 0,005$ po satu, te je na ovom projektu potrošeno 0,01 eura. Trošak je mali jer se cjevovodi koji su korišteni u ovom projektu vrlo brzo izvršavaju.

Microsoft omogućuje vrlo lako računanje cijene cjelokupnog projekta na svom kalkulatoru cijene za *Azure* servise kojem se pristupa preko web preglednika. Na taj način se dolazi do predikcija o početnom budžetu kojeg je potrebno osigurati.

9. Zaključak

Integracija podataka je ključna za uspješno funkcioniranje skladišta podataka. Proces integracije je kompleksan i zahtijeva precizno planiranje, provjere, provedbu i praćenje. Zbog različitih izvora podataka, različitih formata i različitih standarda potrebno je koristiti različite alate i tehnologije kako bi se osigurala uspješna integracija podataka.

Microsoft Azure platforma daje veliku prednost u integraciji podataka, jer nudi mnoštvo servisa koji se lako mogu međusobno povezati. Korištenjem servisa koje pruža Azure, integracija podataka je vrlo jednostavna jer se svim servisima pristupa preko oblaka. Najvažniji servis kod integracije podataka je *Azure Data Factory* koji omogućuje automatiziranje učitavanja podataka u skladište podataka. Automatizacijom se smanjuje mogućnost pogreške i ubrzava proces. Međutim, integracija podataka je samo prvi korak u stvaranju vrijednosti iz podataka. Nakon što su podaci učitani u skladište podataka, potrebno je raditi analize, što uključuje postavljanje analitičkih upita koji vraćaju rezultate brže nego što bi se to dogodilo kod klasičnih operacijskih baza podataka. Koriste se alati za vizualizaciju podataka koji vrlo razumljivo prikazuju zanimljive informacije koje su ključne za napredak tvrtke na tržištu

Britanski matematičar *Clive Humby* je rekao da su podatci nova nafta, i to čak 2006. godine, danas je svima jasno da je definitivno bio u pravu. Prikupljanje podataka se prakticira za velik broj procesa. Jednom prikupljeni podatci se mogu ponovno koristiti, te se mogu koristiti i za različite svrhe. Google karte su napravljene u svrhu lakše navigacije, danas se ti isti podatci koriste i za samovozeće automobile. Teško je predvidjeti kakvi će podatci biti iskoristivi u budućnosti. S obzirom na malu cijenu skladištenja podataka čuvaju se razno razni podatci, a kako bi ti podatci u budućnosti bili iskoristivi potrebno je osigurati kvalitetan proces integracije podataka.

Popis literature

[1] Arul Kumar, V., Akshayaa, L., Madhumidha, K., Radhika, D., & Ramya Kamatchi, E. (2019). Report Generation using Slowly Changing Dimension. *International Journal of Research in Engineering, Science and Management*.

[2] Awoyelu, I., Omodunbi, T., & Udo, J. (2013). Bridging the Gap in Modern Computing Infrastructures: Issues and Challenges of Data Warehousing and Cloud Computing. *Computer and Information Science*, *7*(1), p33. https://doi.org/10.5539/cis.v7n1p33

[3] *Azure Blob Storage | Microsoft Azure*. (n.d.). Retrieved March 5, 2023, from https://azure.microsoft.com/en-us/products/storage/blobs

[4] *Azure Data Factory Documentation—Azure Data Factory*. (n.d.). Retrieved March 5, 2023, from https://learn.microsoft.com/en-us/azure/data-factory/

[5]Bhardwaj, S., Jain, L., & Jain, S. (2010). *CLOUD COMPUTING: A STUDY OF INFRASTRUCTURE AS A SERVICE (IAAS)*.

[6] Cai, L., & Zhu, Y. (2015). The Challenges of Data Quality and Data Quality Assessment in the Big Data Era. *Data Science Journal*, *14*(0), Article 0. https://doi.org/10.5334/dsj-2015-002

[7] Chaudhary, S., Murala, D. P., & Srivastav, V. K. (2011). *A Critical Review of Data Warehouse*. 10.

[8] Cheong, L. K., & Chang, V. (2007). *The Need for Data Governance: A Case Study*. 11.

[9] *Cloud Data Fusion*. (n.d.). Google Cloud. Retrieved March 6, 2023, from https://cloud.google.com/data-fusion

[10] *Cloud Storage*. (n.d.). Google Cloud. Retrieved March 6, 2023, from https://cloud.google.com/storage

[11] de Bruin, B., & Floridi, L. (2017). The Ethics of Cloud Computing. *Science and Engineering Ethics*, *23*(1), 21–39. https://doi.org/10.1007/s11948-016-9759-0

[12] Doan, A., Halevy, A., & Ives, Z. (2012). *Principles of Data Integration*. Elsevier.

[13] El-Sappagh, S. H. A., Hendawi, A. M. A., & El Bastawissy, A. H. (2011). A proposed model for data warehouse ETL processes. *Journal of King Saud University - Computer and Information Sciences*, *23*(2), 91–104. https://doi.org/10.1016/j.jksuci.2011.05.005

[14] Huynh, T. N., Mangisengi, O., & Tjoa, A. M. (n.d.). *Metadata for Object-Relational Data Warehouse*. 9.

[15] *Infographic: Amazon, Microsoft & Google Dominate Cloud Market*. (2022, December 23). Statista Infographics. https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers

[16] Inmon, W. H. (2002). *Building the Data Warehouse* (Third).

[17] Kimball, R., & Casetra, J. (2004). *The Data Warehouse ETL Toolkit*. Wiley Publishing inc.

[18] Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit, 3rd Edition 2013.pdf* (Third).

[19] Ladley, J. (2019). *Data Governance: How to Design, Deploy, and Sustain an Effective Data Governance Program*. Academic Press.

[20] Patel, J., & Patel, A. (2012). *DATA MODELING TECHNIQUES FOR DATA WAREHOUSE*. *2*, 240–246.

[21] Sarka, D., Lah, M., & Jerkič, G. (2012). *Implementing a Data Warehouse with Microsoft SQL Server 2012*.

[22] Satyanarayana, S. (2011). CLOUD COMPUTING : SAAS. *2011*.

[23] *What is Amazon Redshift?* (n.d.). Sumo Logic. Retrieved March 5, 2023, from https://www.sumologic.com/blog/what-is-amazon-redshift/

[24] *What is Amazon S3? - Amazon Simple Storage Service*. (n.d.). Retrieved March 5, 2023, from https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html

[25] *What is AWS Glue? - AWS Glue*. (n.d.). Retrieved March 5, 2023, from https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html

Popis slika

Popis tablica