

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

**Početak rada s Jezerskim
Skladištem Podataka: Teorija i
Tehnologija**

Ivan Derdić

Voditelj: *Alan Joivić*

Zagreb, svibanj 2023.

SADRŽAJ

1. Uvod	1
2. Teorija	2
2.1. Jezersko Skladište Podataka	2
2.2. Sloj Unosa Podataka	2
2.3. Sloj Pohrane Podataka	3
2.4. Sloj Obrade Podataka	4
2.5. Arhitektura i tok podataka	4
3. Tehnologija	6
3.1. Spark	6
3.2. Delta Lake	6
4. Primjer	8
4.1. Opis zadatka	8
4.2. Unos podataka u brončani sloj	9
4.3. Unos podataka u srebrni sloj	10
4.4. Unos podataka u zlatni sloj	12
5. Zaključak	14
6. Literatura	15

1. Uvod

U radu je opisan početak rada s Jezerskim Skladištem Podataka. Rad je napisan s gledišta podatkovnih inženjera, a ne korisnika Jezerskog Skladišta Podataka. U radu se daje definicija Jezerskog Skladišta Podataka, opisuje se njegova arhitektura i opisuje se postupak izrade. Obrađen je reducirani model Jezerskog Skladišta Podataka. Reducirani model je odabran zbog svoje jednostavnosti i zbog toga što je dovoljan za razumijevanje osnovnih principa rada s Jezerskim Skladištem Podataka sa strane podatkovnih inženjera. Cilj rada je opisati osnovne principe rada s Jezerskim Skladištem Podataka i dati osnovu za daljnje istraživanje.

U poglavlju (2) je opisana definicija Jezerskog Skladišta Podataka, navedeni njegovi slojevi i opisana njegova arhitektura. U poglavlju (3) je opisana tehnologija koja je korištena za ostvarivanje Jezerskog Skladišta Podataka. U poglavlju (4) je opisan primjer izrade Jezerskog Skladišta Podataka i dimenzijskog modela podataka.

2. Teorija

U ovom poglavlju se obrađuje teorija Jezerskog Skladišta Podataka. Ne obrađuje se cijela arhitektura nego samo odabrani slojevi te se daje definicija samog Jezerskog Skladišta Podataka. Odabrani slojevi su:

- Sloj Unosa Podataka,
- Sloj Pohrane Podataka,
- Sloj Obrade Podataka.

Odabirom određenih slojeva, u ovom radu se obrađuje samo dio arhitekture Jezerskog Skladišta Podataka, ali se obrađuje teorija koja je potrebna za razumijevanje tehnologije koja se koristi u praktičnom dijelu rada.

2.1. Jezersko Skladište Podataka

Jezersko Skladište Podataka (eng. Data Lakehouse) je arhitektura skladištenja podataka koja kombinira karakteristike Jezera Podataka (eng. Data Lake) i Skladišta Podataka (eng. Data Warehousea). To je centralizirano skladište podataka koje omogućava analizu velike količine strukturiranih i nestrukturiranih vrsta podataka u realnom vremenu ili u kasnijem trenutku. Jezersko skladište podataka omogućuje integraciju podataka iz različitih izvora, olakšava upravljanje podacima, smanjuje troškove i vrijeme potrebno za pripremu podataka za analizu. Ova arhitektura skladištenja podataka postaje sve popularnija u posljednje vrijeme jer olakšava analizu podataka, izvješćavanje i donošenje odluka u stvarnom vremenu. Za detaljniji opis Jezerskog Skladišta Podataka vidjeti [1].

2.2. Sloj Unosa Podataka

Sloj Unosa Podataka (engl. Data Ingestion Layer) u Jezerskom Skladištu Podataka je sloj koji se koristi za prikupljanje i spremanje podataka iz različitih izvora u Jezersko

Skladište Podataka. Ovaj sloj obuhvaća dva načina prikupljanja podataka:

1. serijsko prikupljanje podataka,
2. strujno prikupljanje podataka.

Sloj Unosa Podataka omogućuje podatkovnim inženjerima da učinkovito prikupe podatke iz različitih izvora i formata, poput baza podataka, datoteka ili senzorskih uređaja, te ih jednostavno učitaju u Jezersko Skladište Podataka. Ovaj sloj obično uključuje alate za obradu velikih količina podataka, poput Apache Spark-a, kako bi se omogućilo brzo i učinkovito prikupljanje i spremanje velikih količina podataka. Za detaljni opis Sloja Unosa Podataka vidjeti [1].

2.3. Sloj Pohrane Podataka

U Jezerskom Skladištu Podataka, Sloj Pohrane Podataka obuhvaća skup tehnologija i alata za pohranu velikih količina podataka u različitim formatima, kao što su Apache Hadoop Distributed File System (HDFS), Amazon S3, Azure Blob Storage i Google Cloud Storage.

Osim toga, Delta Lake tehnologija može se koristiti kao sloj pohrane podataka u Jezerskom Skladištu, jer omogućuje verzioniranje podataka, upravljanje transakcijama i omogućuje pohranu podataka u strukturiranom obliku, čime se olakšava proces analize.

Podaci u Sloju Pohrane Podataka se dijele na 3 razine:

- **Sirovi podaci/Brončani sloj** - podaci koji su prikupljeni iz izvora podataka,
- **Obradeni podaci/Srebrni sloj** - podaci koji su transformirani i pripremljeni za analizu,
- **Analizirani podaci/Zlatni sloj** - podaci koji su analizirani i spremljeni u agregiranom obliku.

Sloj pohrane podataka u Jezerskom Skladištu Podataka mora biti dizajniran i konfiguriran na način koji omogućuje brzi i jednostavan pristup podacima za analizu, uz osiguravanje pouzdanosti, sigurnosti i skalabilnosti skladišta. Za detaljni opis Sloja Pohrane Podataka vidjeti [1].

2.4. Sloj Obrade Podataka

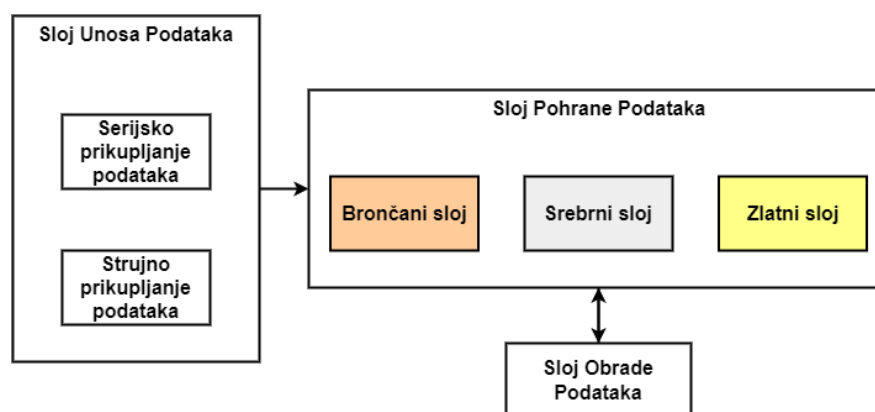
Sloj Obrade Podataka (eng. Data Processing Layer) u Jezerskom Skladištu Podataka odnosi se na skup tehnologija i alata koji omogućuju obradu velikih količina podataka pohranjenih u Jezerskom Skladištu Podataka. Ovaj sloj obično uključuje distribuirane obradne okvire, poput Apache Sparka, Apache Flinka i Apache Beam-a.

Sloj Obrade Podataka je dizajniran kako bi omogućio izvođenje različitih operacija na podacima, uključujući čišćenje, transformiranje, spajanje i agregiranje. Ovaj sloj omogućuje korisnicima da lako i učinkovito izvode složenu obradu velikih skupova podataka, koristeći alate za distribuiranu obradu.

Sloj Obrade Podataka u Jezerskom Skladištu Podataka igra ključnu ulogu u omogućavanju pouzdane, skalabilne i brze obrade podataka pohranjenih u Jezerskom Skladištu Podataka, što omogućuje korisnicima da izvuku vrijednost iz podataka i donose informirane poslovne odluke. Za detaljni opis Sloja Obrade Podataka vidjeti [1].

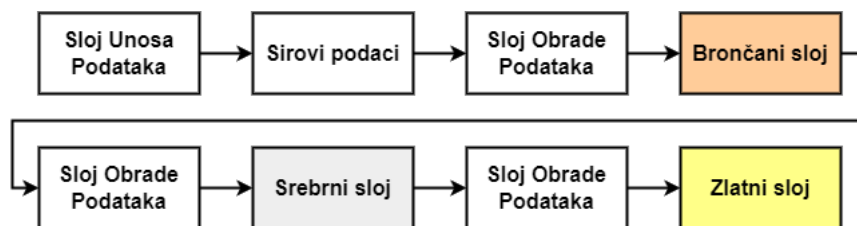
2.5. Arhitektura i tok podataka

Arhitektura Jezerskog Skladišta Podataka prikazana je na slici (2.1). Ona proizlazi iz slojeva opisanih u poglavljima (2.2), (2.3) i (2.4). Sloj Unosa Podataka samo upisuje podatke u Sloj Pohrane Podataka, dok Sloj Obrade Podataka čita i piše podatke u Sloj Pohrane Podataka. Sloj Obrade Podataka čita i piše podatke jer u tom sloju čišćenjem, transformiranjem, spajanjem i agregiranjem podataka ostvaruju podslojevi (Brončani, Srebrni i Zlatni Sloj) Sloja Pohrane Podataka.



Slika 2.1: Arhitektura modela Jezerskog Skladišta Podataka.

Sa slike (2.2) se vidi da postoji tok podataka između Slojeva Unosa, Pohrane i Obrade Podataka. Sloj Unosa Podataka upisuje podatke iz vanjske okoline (baze po-



Slika 2.2: Tok podataka u Jezerskom Skladištu Podataka.

dataka, SFTP serveri, Jezera Podataka) u Sloj Pohrane Podataka. Unosom podataka se dobivaju Sirovi podaci koje Sloj Obrade Podataka dohvaća i unosi u Brončani sloj s najmanjim mogućim brojem transformacija. Sljedeće, Sloj Obrade Podataka dohvaća podatke iz Brončanog sloja te ih čisti, transformira i spaja. Obrađeni podaci Brončanog sloja se unose u Srebrni sloj. Na kraju, Sloj Obrade Podataka dohvaća podatke iz Srebrnog sloja te ih agregira i unosi u Zlatni sloj.

3. Tehnologija

U ovom poglavlju se obrađuju tehnologije koje se koriste za ostvarivanje slojeva Jezerskog Skladišta Podataka u ovom radu. Jezersko Skladište Podataka ne definira tehnologije koje se koriste za ostvarivanje slojeva, već samo slojeve i njihove funkcionalnosti. U ovom poglavlju opisane su sljedeće tehnologije:

- Apache Spark,
- Delta Lake.

3.1. Spark

Spark je distribuirani okvir za obradu podataka otvorenog koda koji je dizajniran za brzu i jednostavnu obradu velikih količina podataka. Spark omogućava programerima da razviju složene aplikacije za obradu podataka i analizu u nekoliko programskih jezika, uključujući Java, Python, Scala i R.

Spark je popularan zbog svoje sposobnosti da brzo i jednostavno obradi velike količine podataka izvršavanjem na grozdu. Osim toga, Spark pruža mnoge biblioteke i alate za obradu podataka, uključujući Spark SQL, Spark Streaming, MLlib i GraphX.

Spark se često koristi u industriji za obradu podataka u stvarnom vremenu, strojno učenje, obradu teksta, obradu slika i druga područja primjene. Spark je postao popularan zbog svoje brzine obrade podataka, skalabilnosti i fleksibilnosti u korištenju.

Spark se u Jezerskom Skladištu Podataka može koristiti za ostvarivanje Sloja Unosa Podataka (vidi poglavlje (2.2)) i Sloja Obrade Podataka (vidi poglavlje (2.4)). Za detaljniji opis Spark-a vidjeti poglavlje 1 iz [2].

3.2. Delta Lake

Delta Lake je open source projekt koji se temelji na Apache Spark-u, a namijenjen je upravljanju i obradi podataka u velikim i složenim analitičkim aplikacijama. Delta

Lake kombinira karakteristike Data Lake-a i skladišta podataka (Data Warehouse-a) u jednoj platformi koja je skalabilna, otporna na pogreške i sposobna za rad u realnom vremenu.

Delta Lake omogućuje pohranu podataka u obliku tabela, s podrškom za transakcije i verzioniranje. To omogućuje korisnicima da jednostavno dodaju, ažuriraju ili brišu podatke, dok se istovremeno održava povijest promjena.

Delta Lake također pruža podršku za naprednu obradu podataka, poput upravljanja s vremenom, verzioniranja shema i upravljanja sinkronizacijom podataka. Ove značajke olakšavaju integraciju s drugim alatima i aplikacijama, što je korisno u velikim poduzećima s kompleksnim IT okruženjima.

Delta Lake se često koristi u poslovima koji zahtijevaju brzu obradu podataka u stvarnom vremenu i velike količine podataka, poput bankarstva, telekomunikacija, e-trgovine i drugih industrija koje se bave velikim količinama podataka.

Delta Lake je tehnologija za upravljanje i obradu podataka koja se može koristiti u sklopu Jezerskog Skladišta Podataka. Može se koristiti za ostvarivanje Sloja Pohrane Podataka (vidi poglavlje (2.3)). Za detaljniji opis Delta Lake-a vidjeti [3]

4. Primjer

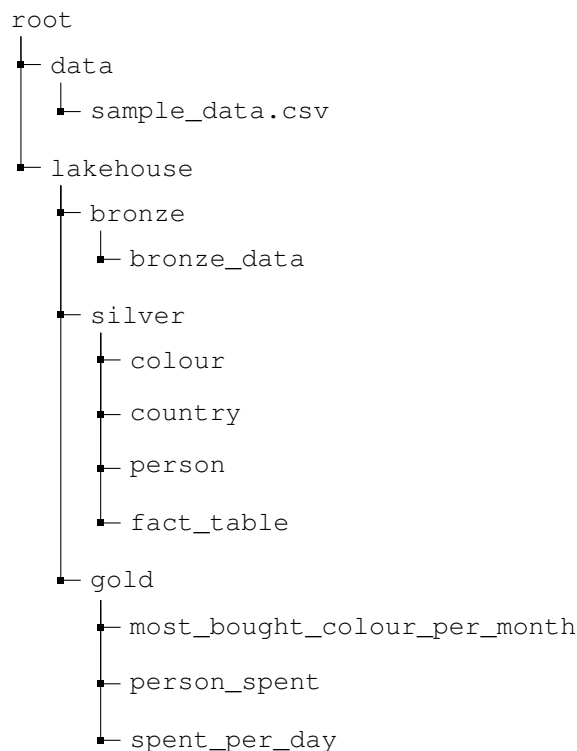
U ovom poglavlju prikazan je primjer izrade Jezerskog Skladišta Podataka i toka podatak definiranog u poglavlju (2.5). Za ostvarivanje Sloja Obrade Podataka korišten je Apache Spark (poglavlje (3.1)), a za ostvarivanje Sloja Skladištenja Podataka korišten je Delta Lake (poglavlje (3.2)). Za ostvarivanje Sloja Unosa Podataka mogao se koristiti Apache Airflow, Azure Data Factory ili neki slični alat. U ovom primjeru nije realiziran Sloj Unosa Podataka pošto bi se sveo na kopiranje podataka iz jednog direktorija u drugi. U nastavku je opisan postupak izrade Jezerskog Skladišta Podataka i dimenzijskog modela podataka.

4.1. Opis zadatka

Kreirati Jezersko Skladište Podataka i dimenzijski model podataka za analizu iz podataka sa sljedećom strukturom:

date datum,
name tekst,
phone tekst,
email tekst,
country tekst,
colour tekst,
currency tekst.

Dana struktura podataka je struktura sirovih podataka. Sirovi podaci su kreirani pomoću stranice generatedata.com. Sirovi podaci su u CSV formatu i nalaze se u datoteci `sample_data.csv` u direktoriju `data` kako je prikazano na slici (4.1). Potrebno je kreirati brončani, srebrni i zlatni sloj Jezerskog Skladišta Podataka. U brončani sloj



Slika 4.1: Struktura direktorija i datoteka u primjeru jednostavnog Jezerskog Skladišta Podataka.

treba učitati sirove podatke, kojima je su dodana polja 'batch_date' i 'input_file'. U srebrni sloj treba učitati podatke iz brončanog sloja i kreirati dimenzijske tablice 'colour', 'country' i 'person' i činjeničnu tablicu 'fact_table'. U zlatni sloj treba učitati podatke iz srebrnog sloja i kreirati činjenične tablice 'most_bought_colour_per_month', 'person_spent' i 'spent_per_day'. Također, za tablice 'spent_per_day' treba kreirati i vizualizaciju podataka. Konačno Jezersko Skladište Podataka bi trebalo imati strukturu direktorija i datoteka prikazano na slici (4.1). U nastavku je opisan postupak izrade Jezerskog Skladišta Podataka i dimenzijskog modela podataka.

4.2. Unos podataka u brončani sloj

Programski kod (4.1) prikazuje dio skripte koja stvara tablicu 'bronze_data' u Jezerskom Skladištu Podataka. Za čitanje sirovih podataka koristi se funkcija `read_raw_data` koja koristi Sparkov `DataFrameReader` klasu za čitanje podataka iz datoteke `sample_data.csv`. Nakon čitanja sirovih podataka dodaju se polja 'batch_date' i 'input_file' koja se koriste za praćenje podataka kroz Jezersko Skladište Podataka. Polje 'batch_date' sadrži datum kada su podaci učitani u Jezersko Skladište Podataka, a polje 'input_file' sadrži

naziv datoteke iz koje su podaci učitani. Nakon dodavanja polja podaci se zapisuju u Jezersko Skladište Podataka u tablicu ‘bronze_data’.

Programski kod 4.1: Dio skripte koja stvara tablicu ‘bronze_data’ u Jezerskom Skladištu Podataka.

```
raw_df = read_raw_data(  
    path="data/sample_data.csv",  
    schema=schema,  
    spark=spark  
)  
  
# Add batch date and input file name to raw data  
batch_date = datetime.now().strftime("%Y-%m-%d")  
raw_df = raw_df.withColumn(  
    "batch_date",  
    to_date(  
        lit(batch_date),  
        "yyyy-MM-dd"  
    )  
)  
  
raw_df = raw_df.withColumn(  
    "input_file",  
    input_file_name()  
)  
  
# Write raw data to lakehouse bronze layer  
write_to_lakehouse(  
    df=raw_df,  
    path="lakehouse/bronze/bronze_data",  
    partition=True  
)
```

4.3. Unos podataka u srebrni sloj

Kod unosa podatak u srebrni sloj potrebno je kreirati dimenzijske tablice i činjeničnu tablicu. Programski kod (4.2) prikazuje dio skripte koja stvara tablicu ‘fact_table’ u Jezerskom Skladištu Podataka. Za kreiranje tablice ‘fact_table’ potrebno je učitati podatke iz tablice ‘bronze_data’ i kreirati dimenzijske tablice ‘person’, ‘country’ i ‘colour’. Dimenzijske tablice se kreiraju tako da se iz tablice ‘bronze_data’ izvuku potrebni stupci i uklone duplikati. Nakon toga se dimenzijskim tablicama dodaju suro-

gatni ključevi. Surogatni ključevi se dodaju tako da se dimenzijske tablice mogu spajati s činjeničnom tablicom samo preko jednog stupca. Nakon dodavanja surogatnih ključeva dimenzijske tablice se spajaju s tablicom 'bronze_data'. Iz tablice dobivene spajanjem se odabiru samo potrebni stupci. Potrebni stupci su surogatni ključevi dimenzijskih tablica, datumski stupci i mjere. Nakon odabira stupaca tablica se zapisuje u Jezersko Skladište Podataka u tablicu 'fact_table'. Opisani postupak za kreiranje činjenične tablice prikazan je u programskom kodu (4.2).

Programski kod 4.2: Dio skripte koja stvara tablicu 'fact_table' u Jezerskom Skladištu Podataka.

```
fact_df = bronze_df.join(
    person_df,
    [
        df.first_name == person_df.first_name,
        df.last_name == person_df.last_name,
        df.phone == person_df.phone,
        df.email == person_df.email,
        df.country == person_df.country
    ],
    how='inner'
)

fact_df = fact_df.join(
    country_df,
    'country',
    how='inner'
)

fact_df = fact_df.join(
    colour_df,
    'colour',
    how='inner'
)

fact_df = fact_df.select(
    'date',
    'person_id',
    'country_id',
    'colour_id',
    'currency'
)
```

```

write_to_lakehouse(
    df=fact_df,
    path="lakehouse/silver/fact_table"
)

```

4.4. Unos podataka u zlatni sloj

Kod unosa podataka u zlatni sloj potrebno je kreirati tablice ‘most_bought_colour_per_month’, ‘person_spent’ i ‘spent_per_day’. Programski kod (4.3) prikazuje dio skripte koja stvara tablicu ‘spent_per_day’ u Jezerskom Skladištu Podataka. Za kreiranje tablice ‘spent_per_day’ potrebno je učitati podatke iz tablice ‘fact_table’ i izračunati ukupan iznos potrošen po danu. Nakon izračuna ukupnog iznosa potrošenog po danu tablica se zapisuje u Jezersko Skladište Podataka u tablicu ‘spent_per_day’. Za tablice ‘most_bought_colour_per_month’ i ‘person_spent’ postupak je sličan kao i za tablicu ‘spent_per_day’. Postupak se razlikuje po načinu računanju mjera i potrebnih dodatnih dimenzijskih tablica. Na slici (4.2) se vidi graf koji prikazuje podatke iz tablice ‘spent_per_day’. Na grafu se prikazuje zadnjih deset dana. Na x-osi se nalazi datum, a na y-osi se nalazi ukupan iznos potrošen po danu.

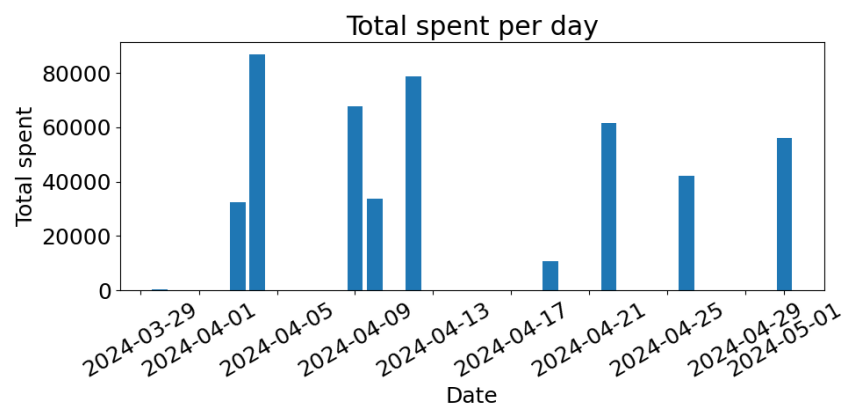
Programski kod 4.3: Dio skripte koja stvara tablicu ‘spent_per_day’ u Jezerskom Skladištu Podataka.

```

spent_per_day_df = (
    fact_df
    .groupBy('date')
    .sum('currency')
    .withColumnRenamed(
        'sum(currency)',
        'spent_per_day'
    )
)

write_to_lakehouse(
    df=spent_per_day_df,
    path="lakehouse/gold/spent_per_day"
)

```



Slika 4.2: Graf koji prikazuje podatke iz tablice 'spent_per_day'.

5. Zaključak

Rezultat rada je poznavanje osnovnih principa rada s Jezerskim Skladištem Podataka. U radu je opisana definicija Jezerskog Skladišta Podataka, navedeni njegovi slojevi i opisana njegova arhitektura. U radu je opisana tehnologija koja je korištena za ostvarivanje Jezerskog Skladišta Podataka. Kroz primjer je dan opis visoke razine izrade Jezerskog skladišta podataka. Cilj rada je postignut.

Najveći problem u izradi rada je bilo postavljanje okoline za izradu primjera. Spark je kompleksan alat za postaviti.

Daljnji koraci bi bili istraživanje cijelog modela Jezerskog Skladišta Podataka i istražiti neke druge tehnologije koje je moguće koristiti za ostvarivanje Jezerskog Skladišta Podataka.

6. Literatura

- [1] P. Menon, *Data Lakehouse in Action*. Packt Publishing Pvt Ltd, 2022.
- [2] J. S. Damji, B. Wenig, T. Das, and D. Lee, *Learning Spark: Lightning-Fast Data Analytics*. O'Reilly Media, 2020.
- [3] Microsoft, “What is delta lake?,” 2023.

Početak rada s Jezerskim Skladištem Podataka: Teorija i Tehnologija

Sažetak

U radu je opisana teorija i tehnologija Jezerskog Skladišta Podataka. Rad je napisan s gledišta podatkovnih inženjera. U radu je dan primjer izrade jednostavnog Jezerskog Skladišta Podataka.

Ključne riječi: Jezersko Skladište Podataka, Spark, Delta Lake, Dimenzijski model podataka