SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Početak rada s jezerskim skladištem podataka: Teorija i Tehnologija

Ivan Derdić
Voditelj: *izv. prof. dr. sc. Alan Jović*

Zagreb, svibanj 2023.

SADRŽAJ

1. Uvod

U radu su opisane osnove 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) dana je definicija jezerskog skladišta podataka, navedeni njegovi slojevi i opisana njegova arhitektura. U poglavlju (3) opisana je tehnologija koja je korištena za ostvarivanje jezerskog skladišta podataka. U poglavlju (4) opisan je 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 je arhitektura skladištenja podataka koja kombinira karakteristike jezera podataka (eng. Data Lake) i skladišta podataka (eng. Data Warehouse). 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, izvještavanje 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 tri razine:

- **sirovi podaci / brončani sloj** - podaci koji su prikupljeni iz izvora podataka,
- **obrađeni 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 oblikovan 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

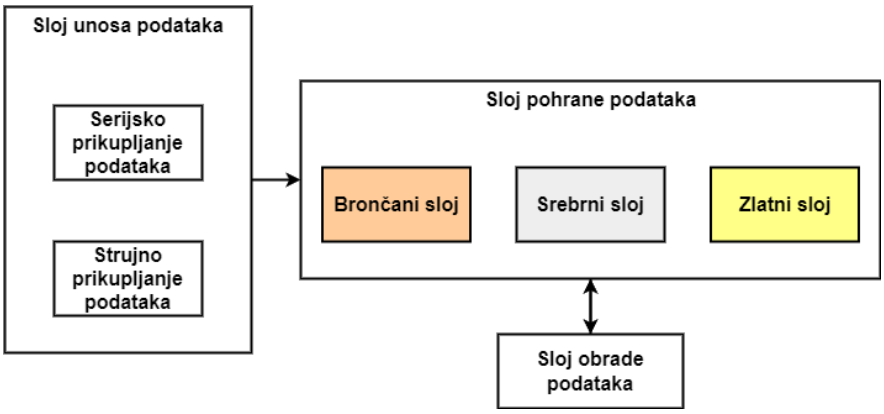
pohranjenih u jezerskom skladištu podataka. Ovaj sloj obično uključuje distribuirane obradne okvire, poput Apache Sparka, Apache Flinka i Apache Beama.

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].

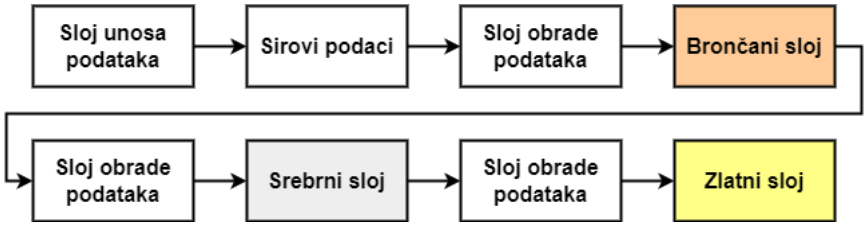
2.5. Arhitektura i tok podataka

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 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 agregiranje podataka ostvaruju podslojevi (brončani, wrebrni 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 podataka, 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



Slika 2.2: Tok podataka u jezerskom skladištu podataka.

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 Sparka vidjeti poglavlje 1 iz [2].

3.2. Delta Lake

Delta Lake je projekt otvorenog koda koji se temelji na Apache Sparku, a namijenjen je upravljanju i obradi podataka u velikim i složenim analitičkim aplikacijama. Delta

Lake kombinira karakteristike jezera podataka i skladišta podataka u jednoj platformi koja je skalabilna, otporna na pogreške i sposobna za rad u realnom vremenu.

Delta Lake omogućuje pohranu podataka u obliku tablica, 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 podataka 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 podata mogao se koristiti Apache Airflow, Azure Data Factory ili neki slični alat. U ovom primjeru nije realiziran sloj unosa podataka jer 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

Izraditi jezersko skladište podataka i dimenzijski model podataka (vidjeti [4]) za analizu iz podataka sa sljedećom strukturom:

date datum,

name tekst,

phone tekst,

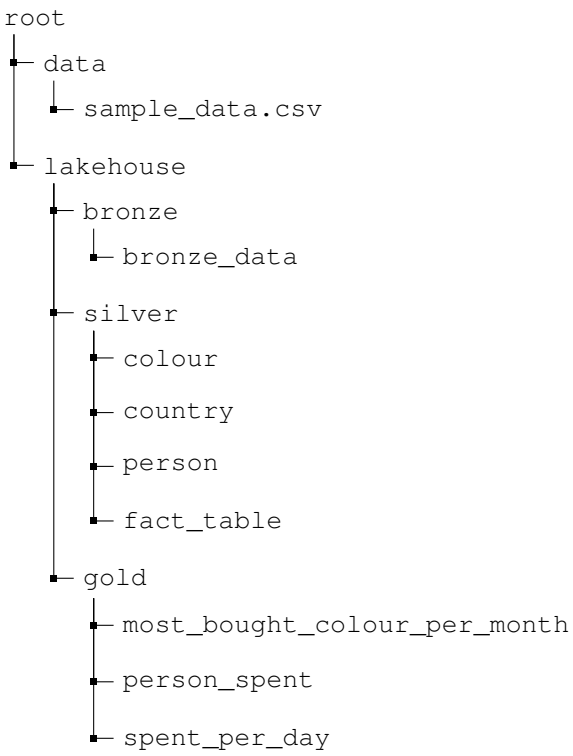
email tekst,

country tekst,

colour tekst,

currency tekst.

Dana struktura podataka je struktura sirovih podataka. Sirovi podaci su generirani pomoću stranice *generatedata.com*. Sirovi podaci su u formatu CSV i nalaze se u datoteci `sample_data.csv` u direktoriju `data` kako je prikazano na slici (4.1). Potrebno je izraditi 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 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 napraviti 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 gradi 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 gradi tablicu ‘bronze_data’ u jezerskom skladištu podataka [5].

```
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 podataka u srebrni sloj potrebno je napraviti dimenzijske tablice i činjeničnu tablicu. Programski kod (4.2) prikazuje dio skripte koja gradi tablicu ‘fact_table’ u jezerskom skladištu podataka. Za izradu tablice ‘fact_table’ potrebno je učitati podatke iz tablice ‘bronze_data’ i kreirati dimenzijske tablice ‘person’, ‘country’ i ‘colour’. Dimenzijske tablice se grade tako da se iz tablice ‘bronze_data’ izvuku potrebni stupci i uklone duplikati. Nakon toga se dimenzijskim tablicama dodaju surogatni 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 izradi je u programskom kodu (4.2).

Programski kod 4.2: Dio skripte koja stvara tablicu 'fact_table' u jezerskom skladištu podataka [5].

```
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"
)
```

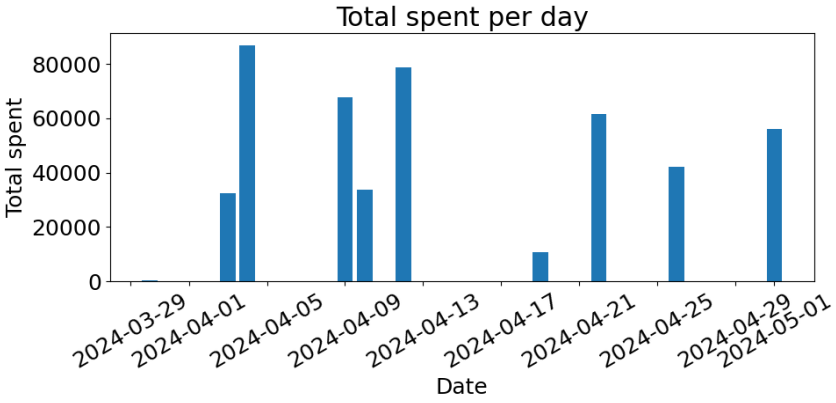
4.4. Unos podataka u zlatni sloj

Kod unosa podataka u zlatni sloj potrebno je napraviti tablice ‘most_bought_colour_per_month’, ‘person_spent’ i ‘spent_per_day’. Programski kod (4.3) prikazuje dio skripte koja gradi tablicu ‘spent_per_day’ u jezerskom skladištu podataka. Za izradu 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 ralikuje 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 gradi tablicu ‘spent_per_day’ u jezerskom skladištu podataka [5].

```
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 istraživanije 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?." https://learn.microsoft.com/en-us/azure/databricks/delta/, 2023.

[4] "What is Dimensional Modeling in Data Warehouse? Learn Types." https://www.guru99.com/dimensional-model-data-warehouse.html, 2023.

[5] I. Derdić, "Simple Lakehouse." simple_lakehouse.py, 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