Izveštaj o realizaciji projekta: Analiza letova u MongoDB-u

Oktobar 2025.

Contents

1 Uvod

Ovaj izveštaj sažima ključne elemente za realizaciju projekta analize podataka o letovima u MongoDB-u, baziran na zahtevima za predispitnu obavezu. Projekt koristi podatke iz CSV fajlova (`US_flights_2023.csv`, `Cancelled_Diverted_2023.csv`, `airports_geolocation.cs` `weather_meteo_by_airport.csv`, `airports.csv`, `airport_frequencies.csv`, `runways.csv`) i uključuje pripremu podataka, MongoDB upite, optimizaciju performansi, i vizualizaciju rezultata. Izveštaj pokriva sedam faza projekta, sa fokusom na realno stanje podataka i praktičnu implementaciju.

2 Inicijalna priprema skupa podataka

Podaci su učitani iz CSV fajlova koristeći Python biblioteku `pymongo`. Strukturirani su prema sledećoj logičkoj šemi:

- **flights**: Podaci o letovima (`FlightDate`, `Airline`, `Dep_Airport`, `Dep_Delay`, `Delay_Weather`, itd.).
- **airports**: Informacije o aerodromima (`iata_code`, `ident`, `type`, `elevation_ft`).
- **weather**: Vremenski podaci (`airport_id`, `time`, `prcp`, `wspd`).
- **frequencies**: Frekvencije aerodroma (`airport_ident`, `type`, `frequency_mhz`).
- **runways**: Podaci o pistama (`airport_ident`, `length_ft`, `surface`).
- **geolocation**: Geolokacija aerodroma (`IATA_CODE`, `CITY`, `LATITUDE`).

Python kod za učitavanje podataka:

```python
import pandas as pd
import pymongo
import os
client = pymongo.MongoClient("mongodb://localhost:27017/")
db = client["flights_db"]
folder = r"C:\Users\tijan\Desktop\sbp"
files = {
    "US_flights_2023": "US_flights_2023.csv",
    "Cancelled_Diverted_2023": "Cancelled_Diverted_2023.csv",
    "Airports_Geolocation": "airports_geolocation.csv",
    "Weather_Meteo_by_Airport": "weather_meteo_by_airport.csv",
    "Airports": "airports.csv",
    "Airport_frequencies": "airport_frequencies.csv",
    "Runways": "runways.csv"
}
for name, filename in files.items():
    df = pd.read_csv(os.path.join(folder, filename))
    df = df.where(pd.notnull(df), None)
    collection = db[name.lower()]
    collection.insert_many(df.to_dict("records"))
```

Preporuke: Testiraj učitavanje sa manjim skupom podataka (npr. `nrows=10000`) zbog velikih fajlova (6.7M redova u `US_flights_2023.csv`). Proveri nedostajuće vrednosti i osiguraj da MongoDB server radi.

3 Analiza skupa podataka

Deset kompleksnih agregacionih upita je razvijeno za analizu kašnjenja letova, povezujući kolekcije preko `$lookup`, `$unwind`, `$match`, `$group`, i `$sort`. Primer upita (aerodromi sa kašnjenjem > 15 minuta, padavine > 10 mm, piste > 10000 ft):

```
db.flights.aggregate([
    {"$match": {"departure.delay": {"$gt": 15}}},
    {"$lookup": {
        "from": "weather", "localField": "departure.airport",
        "foreignField": "airport_id", "as": "weather_info"
    }},
    {"$unwind": "$weather_info"},
    {"$match": {"weather_info.weather.prcp": {"$gt": 10}}},
    {"$lookup": {
        "from": "airports", "localField": "departure.airport",
        "foreignField": "iata_code", "as": "airport_info"
    }},
    {"$unwind": "$airport_info"},
    {"$match": {"airport_info.type": "large_airport"}},
    {"$lookup": {
        "from": "runways", "localField": "airport_info.ident",
        "foreignField": "airport_ident", "as": "runway_info"
    }},
    {"$unwind": "$runway_info"},
    {"$match": {"runway_info.length_ft": {"$gt": 10000}}},
    {"$group": {
        "_id": "$departure.airport",
        "avg_delay": {"$avg": "$departure.delay"},
        "count": {"$sum": 1}
    }},
    {"$sort": {"avg_delay": -1}},
    {"$limit": 5}
])
```

Preporuke: Pokreni upite u MongoDB Compass-u ili Python-u. Proveri rezultate i osiguraj da sve kolekcije imaju podatke.

4 Analiza performansi upita

Performanse upita su analizirane koristeći `explain()`. Uska grla uključuju:

- **Puni scanovi**: Bez indeksa na poljima poput `departure.airport`, upiti skeniraju celu kolekciju.
- **Spori $lookup**: Velike kolekcije (`runways`: 41k redova) usporavaju povezivanje.
- **Veliki rezultati**: `$unwind` generiše mnogo dokumenata.

Primer merenja vremena u Python-u:

```
import time
start = time.time()
result = list(db.flights.aggregate(pipeline))
end = time.time()
print(f"Vreme izvravanja: {end - start} sekundi")
```

Preporuke: Koristi `explain("executionStats")` za detalje o scan-ovima. Izmeri vreme za svaki upit (npr. 10 ponavljanja).

5 Optimizacija performansi upita

Optimizacija uključuje:

- **Indeksiranje**: Kreiranje indeksa na ključnim poljima.
- **Restrukturiranje šeme**: Denormalizacija (npr. ugnežđavanje `weather_summary` u `flights`).

Primer indeksiranja:

```
db.flights.create_index([("departure.airport", 1)])
db.airports.create_index([("iata_code", 1), ("ident", 1)])
db.weather.create_index([("airport_id", 1), ("time", 1)])
```

Preporuke: Testiraj performanse nakon indeksiranja. Denormalizuj često korišćene podatke (npr. `runway_length` u `airports`).

6 Prerada upita

Upiti su prilagođeni za novu šemu. Primer: Umesto `$lookup` za `weather`, koristi se `departure.weather.prcp`.

```
db.flights.aggregate([
    {"$match": {
        "departure.delay": {"$gt": 15},
        "departure.weather.prcp": {"$gt": 10},
        "departure.airport_type": "large_airport",
        "departure.runway_length": {"$gt": 10000}
    }},
    {"$group": {
        "_id": "$departure.airport",
        "avg_delay": {"$avg": "$departure.delay"},
        "count": {"$sum": 1}
    }},
    {"$sort": {"avg_delay": -1}},
    {"$limit": 5}
])
```

Preporuke: Proveri kompatibilnost upita sa novom šemom u Compass-u.

7 Uporedna analiza performansi

Performanse su upoređene pre i posle optimizacije:

- **Pre**: COLLSCAN, sporo (npr. 2.5s za upit 1).
- **Posle**: IXSCAN, brzo (npr. 0.5s za upit 1).

Primer za dijagram:

```
import matplotlib.pyplot as plt
queries = ["Upit 1", "Upit 2", "Upit 3"]
before = [2.5, 3.0, 2.8]
after = [0.5, 0.7, 0.6]
plt.bar(queries, before, width=0.4, label="Pre optimizacije")
```

```
plt.bar([q + 0.4 for q in range(len(queries))], after, width=0.4, label="
    Posle optimizacije")
plt.xlabel("Upiti")
plt.ylabel("Vreme (s)")
plt.legend()
plt.savefig("performance_comparison.png")
```

Preporuke: Uporedi `executionTimeMillis` iz `explain()`. Vizualizuj rezultate koristeći `matplotlib`.

8 Vizualizacija rezultata

Rezultati upita su vizualizovani u Metabase-u:

- **Podešavanje**: Poveži Metabase sa MongoDB (`mongodb://localhost:27017/flights`$_{db}$).**Dashboard**
  $Barchart - ovizakanjenja, piechart - ovizaavio - kompanije.$
  Primer: Bar chart prosečnih kašnjenja po aerodromima za upit 1.
  Preporuke: Eksportuj grafike iz Metabase-a kao PNG/JPG za prezentaciju.

9 Zaključak

Ovaj izveštaj pruža osnovu za realizaciju projekta, uključujući učitavanje podataka, kompleksne upite, optimizaciju, i vizualizaciju. Sledeći koraci:

- Testiraj učitavanje podataka i upite.
- Primeni indekse i denormalizaciju.
- Vizualizuj rezultate u Metabase-u.