Leonard Puškáč

PDT - Zadanie #5 - Elasticsearch

Úloha 1

Na vytvorenie troch inštancií elasticu som použil docker.
Najprv som si stiahol docker image s príkazom:

```
docker pull docker.elastic.co/elasticsearch/elasticsearch:8.5.3
```

Potom som si vytvoril docker-compose.yml súbor podľa stránky https://www.elastic.co/guide/en/elasticsearch/reference/current/docker.html a vytvoril kontainery príkazom:

```
docker compose up
```

Toto vytvorilo jeden kontainer so štyrmi pod-kontainermi - jeden pre kibana a tri pre elastic search.

Úloha 2

Vytvorenie Indexu

Na vytvorenie indexu som použil metódu PUT s tým, že som request poslal na port 9200. Tento request som definoval v json.
Počet shardov som nastavil na 3, tak aby sa rovnal počtu nodov - nepotrebujeme viac shardov - takýmto spôsobom každý node má jeden shard. Väčší počet shardov by znamenal, že niektorý z nodov by mal viacero shardov, a musel by pri requestoch na ne spájať výsledky zo všetkých svojich shardov.
Počet replík som nastavil na 2.

Úloha 3

Pôvodná schéma tabuliek zo zadania 1:
Pôvodná schéma

Podľa tejto schémy som vytvoril mapping zo všetkých tabuľiek. Tento mapping obsahuje properties - každá má názov jednej z tabuľiek, okrem tabuľky conversation references (ta je nahradená property s názvom "parent_tweet"), a okrem context annotations, aby sme sa zbavili čisto relačnej tabuľky - môžme proste uložiť všetky relevantné entities a domains do daného dokumentu.

Tie tabuľky ktoré sú v one-to-many relationship s daným tweetom, sú definované v mappingu ako "nested" property.

Celý post request je uložený v súbore create_mapping.json

Vytvorenie mapping

Úloha 4

Keďže treba pridať analyzéry, index vytvorený v úlohe 2 spolu s mapping som vymazal, a vytvorím nový s rovnakým mappingom (plus analyzery na niektore fields).

Zmazanie indexu

Následne som vytvoril nový index s rovnakým názvom, s tým že som pridal požadované analyzéry, použitím PUT requestu. Pri author.description a tweet_info.description som pridal analyzér englando, a pri hashtags som pridal normalizer na lowercase. author name, a username teraz maju field pre custom_ngram, a name a description majú field pre custom_shingles (name má tým pádom obidva).\

Oproti úlohe 3 som ešte spravil zmenu, čo sa týka conversation_references - uvedomil som si, že tam je treba tiež spraviť nested pretože pre tweet existuje viacero referencií.

Celý request je možné vidieť v súbore create_index_mapping_analysers.json

Vytvorenie nového indexu

Úloha 5

Na denormalizovanie dát z tabuľiek som si vytvoril jeden veľký sql dopyt ktorý spája všetky tabuľky do jednej - tento sql dopyt je v súbore denormalize_tables.sql.

Bohužial som toto zadanie začal robiť veľmi neskoro, a pri zbiehani denormalizacie som zisitl, že nemam dostatočný disk space... takže rip.