FIIT STU

Zadanie 1: Import tweetov do PostgreSQL

Dokumentácia

Meno: Bc. Martin Šváb
Študijný program: Inteligentné softvérové systémy
Ročník: ING 1, cvičenie: piatok 13:00
Predmet: Pokročilé databázové technológie
Cvičiaci: Ing. Ján Balažia, PhD.
Akademický rok: 2022/23

Link na GirHub: FIIT-DBS/zadanie-pdt-MS-101 (github.com)

Opis algoritmu

Program je rozdelený na 4 časti:

1. Vytvorenie prázdnych tabuliek – funkcia „**create_tables()**“
    a. Rovnaké dátové typy a primárne kľúče ako v diagrame zadania
    b. Neobsahujú cudzie kľúče pre zrýchlenie vkladania dát a predídeniu konfliktov (napr. keď vložím conversation_reference, keď ešte nebola vložená rodičovská konverzácia)
2. Spracovanie súboru „authors.jsonl“ – funkcia „**insert_authors()**“
    a. Otvorí sa súbor „authors.jsonl“. Tento sa spracúva sériovo.
    b. Prečíta sa 10 tisíc riadkov zo súboru.
    c. Prečítané riadky sa spracujú - postupne sa vytvára SQL query na vloženie do tabuľky „authors“.
        i. Záznamy sa nevkladajú v jednotlivých sql dotazoch ale hromadne v jednom dotaze naraz na zníženie réžie.
        ii. Id prečítaných záznamov sa ukladajú do hash máp aby sa predišlo vloženiu duplicitných záznamov. Ak sa prečíta záznam s id, ktoré sa nachádza v hash mape, tak je zahodený.
    d. Vykoná sa transakcia.
    e. Vypíše sa časový priebeh v požadovanom formáte do CSV súboru „authors.csv“.
    f. Opakuj krok b až kým sa neprečíta celý súbor.
3. Spracovanie súboru „conversations.jsonl“ – funkcia „**insert_conversations()**“
    a. Totožné s funkciou „insert_authors()“. Číta sa zo súboru „conversations.jsonl“ a časový priebeh sa zapisuje sa do súboru „conversations.csv“.
4. Úprava tabuliek – funkcia „**alter_tables()**“
    a. Vymažú sa záznamy z tabuľky conversation_references, ktorým chýba autor referovanej konverzácie – majú neplatný cudzí kľúč.
    b. Nastavia sa cudzie kľúče pre všetky tabuľky.

Použité technológie:

- Python – tento programovací jazyk som si vybral, lebo sa v ňom jednoducho pracuje. Na vypracovanie tohto zadania postačovalo napísať efektívny skript. Python bol pre toto zadanie najvhodnejšou voľbou.
- PostgreSQL – je to pokročilý databázový systém, ktorý obsahuje množstvo pokročilých funkcionalít.

SQL dotazy

Vytvorenie tabuliek:

Nasledujúce dotazy sa nachádzajú vo funkcii „create_tables()“. Sú vytvárané v takom poradí aby pri odstránení predošlej tabuľky nedochádzalo ku konfliktom s cudzími kľúčmi (napr. tabuľky context_entities a context_domains vymažem až po vymazaní tabuľky context_annotations)

Vytvorenie tabuľky context_annotations	Vytvorenie tabuľky context_domains
DROP TABLE IF EXISTS context_annotations; CREATE TABLE context_annotations(id BIGSERIAL PRIMARY KEY, conversation_id INT8 NOT NULL, context_domain_id INT8 NOT NULL, context_entity_id INT8 NOT NULL);	DROP TABLE IF EXISTS context_domains; CREATE TABLE context_domains(id INT8 PRIMARY KEY, name VARCHAR(255) NOT NULL, description TEXT);

Vytvorenie tabuľky context_entities	Vytvorenie tabuľky conversation_hashtags
DROP TABLE IF EXISTS context_entities; CREATE TABLE context_entities(id INT8 PRIMARY KEY, name VARCHAR(255) NOT NULL, description TEXT);	DROP TABLE IF EXISTS conversation_hashtags; CREATE TABLE conversation_hashtags(id BIGSERIAl PRIMARY KEY, conversation_id INT8, hashtag_id INT8);

Vytvorenie tabuľky hashtags	Vytvorenie tabuľky annotations
DROP TABLE IF EXISTS hashtags; CREATE TABLE hashtags(id BIGSERIAL PRIMARY KEY, tag TEXT UNIQUE);	DROP TABLE IF EXISTS annotations; CREATE TABLE annotations(id BIGSERIAL PRIMARY KEY, conversation_id INT8 NOT NULL, value TEXT NOT NULL, type TEXT NOT NULL, probability numeric(4, 3) NOT NULL);

Vytvorenie tabuľky links	Vytvorenie tabuľky conversation_references
DROP TABLE IF EXISTS links; CREATE TABLE links(id BIGSERIAL PRIMARY KEY, conversation_id INT8 NOT NULL, url VARCHAR(2048) NOT NULL, title TEXT, description TEXT);	DROP TABLE IF EXISTS conversation_references; CREATE TABLE conversation_references(id BIGSERIAl PRIMARY KEY, conversation_id INT8 NOT NULL, parent_id INT8 NOT NULL, type VARCHAR(20) NOT NULL **);**

Vytvorenie tabuľky conversations	Vytvorenie tabuľky authors
DROP TABLE IF EXISTS conversations; CREATE TABLE conversations(id INT8 PRIMARY KEY, author_id INT8 NOT NULL, content TEXT NOT NULL, possibly_sensitive BOOL NOT NULL, language VARCHAR(3) NOT NULL, source TEXT NOT NULL, retweet_count INT4, reply_count INT4, like_count INT4, quote_count INT4, created_at TIMESTAMP WITH TIME ZONE NOT NULL);	DROP TABLE IF EXISTS authors; CREATE TABLE authors(id INT8 PRIMARY KEY, name VARCHAR(255), username VARCHAR(255), description TEXT, followers_count INT4, following_count INT4, tweet_count INT4, listed_count INT4);

Vloženie do tabuliek:

Nasledujúce dotazy sa nachádzajú vo funkciách „insert_authors()“ a „insert_conversations()“. Vkladá sa viac riadkov naraz aby sa zmenšila réžia.

Vloženie do tabuľky authors	Vloženie do tabuľky conversations
INSERT INTO authors (id, name, username, description, followers_count, following_count, tweet_count, listed_count) VALUES (%s, %s, %s, %s, %s, %s, %s, %s), … (%s, %s, %s, %s, %s, %s, %s, %s)	INSERT INTO conversations (id, author_id, content, possibly_sensitive, language, source, retweet_count, reply_count, like_count, quote_count, created_at) VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s), … (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)

Vloženie do tabuľky hashtags	Vloženie do tabuľky conversation_hashtags
INSERT INTO hashtags (tag) VALUES (%s), ... (%s) RETURNING id	INSERT INTO conversation_hashtags (conversation_id, hashtag_id) VALUES (%s, %s) ... (%s, %s)

Vloženie do tabuľky annotations	Vloženie do tabuľky links
INSERT INTO annotations (conversation_id, value, type, probability) VALUES (%s, %s, %s, %s), ... (%s, %s, %s, %s)	INSERT INTO links (conversation_id, url, title, description) VALUES (%s, %s, %s, %s), ... (%s, %s, %s, %s)

Vloženie do tabuľky conversation_references	Vloženie do tabuľky context_annotations
INSERT INTO conversation_references (conversation_id, parent_id, type) VALUES (%s, %s, %s), ... (%s, %s, %s)	INSERT INTO context_annotations(conversation_id, context_domain_id, context_entity_id) VALUES (%s, %s, %s), ... (%s, %s, %s)

Vloženie do tabuľky context_domains	Vloženie do tabuľky context_entities
INSERT INTO context_domains (id, name, description) VALUES (%s, %s, %s), ... (%s, %s, %s)	INSERT INTO context_entities (id, name, description) VALUES (%s, %s, %s), ... (%s, %s, %s)

Úprava tabuliek:

Nasledujúce dotazy sa nachádzajú vo funkcii „alter_tables(). Pomocou prvého dotazu sa vymažú neplatné konverzačné referencie (nemajú platné parent_id) a následne sa nastavia cudzie kľúče v tabuľkách.

Vymazanie neplatných konverzačných referencií

```
DELETE FROM conversation_references
WHERE id IN (
        SELECT conversation_references.id FROM conversation_references
        LEFT JOIN conversations on conversations.id = conversation_references.parent_id
        WHERE conversations.id IS NULL
)
```

Pridanie cudzieho kľúča do tabuľky conversations (author_id -> authors(id))	**Pridanie cudzieho kľúča do tabuľky conversation_hashtags (conversation_id -> conversations(id))**
ALTER TABLE conversations ADD FOREIGN KEY (author_id) REFERENCES authors(id)	ALTER TABLE conversation_hashtags ADD FOREIGN KEY (conversation_id) REFERENCES conversations(id)

Pridanie cudzieho kľúča do tabuľky conversation_hashtags (hashtag_id -> hashtags(id))	**Pridanie cudzieho kľúča do tabuľky conversation_hashtags (conversation_id -> conversations(id))**
ALTER TABLE conversation_hashtags ADD FOREIGN KEY (hashtag_id) REFERENCES hashtags(id)	ALTER TABLE annotations ADD FOREIGN KEY (conversation_id) REFERENCES conversations(id)

Pridanie cudzieho kľúča do tabuľky links (conversation_id -> conversations(id))	**Pridanie cudzieho kľúča do tabuľky conversation_references (conversation_id -> conversations(id))**
ALTER TABLE links ADD FOREIGN KEY (conversation_id) REFERENCES conversations(id)	ALTER TABLE conversation_references ADD FOREIGN KEY (conversation_id) REFERENCES conversations(id)

Pridanie cudzieho kľúča do tabuľky conversation_references (parent_id -> conversations(id))	**Pridanie cudzieho kľúča do tabuľky context_annotations (conversation_id -> conversations(id))**
ALTER TABLE conversation_references ADD FOREIGN KEY (parent_id) REFERENCES conversations(id)	ALTER TABLE context_annotations ADD FOREIGN KEY (conversation_id) REFERENCES conversations(id)

Pridanie cudzieho kľúča do tabuľky context_annotations (context_domain_id -> context_domains(id))	**Pridanie cudzieho kľúča do tabuľky context_annotations (context_entity_id -> context_entities(id))**
ALTER TABLE context_annotations ADD FOREIGN KEY (context_domain_id) REFERENCES context_domains(id)	ALTER TABLE context_annotations ADD FOREIGN KEY (context_entity_id) REFERENCES context_entities(id)

Časový priebeh

Vytvorenie tabuliek: 0m 0s

Spracovanie autorov: 4m 3s

Spracovanie konverzácií: 102m 55s

Úprava tabuliek: 11m 49 s

Časový priebeh spracovávania autorov (každých 10000 záznamov) je v priečinku „output“ v súbore „**authors.csv**“.

Časový priebeh spracovávania konverzácií (každých 10000 záznamov) je v priečinku „output“ súbore „**conversations.csv**“.

Výsledná databáza

Počty záznamov:

```sql
SELECT
    (
        SELECT COUNT(*) FROM annotations
    ) AS annotations_count,
    (
        SELECT COUNT(*) FROM authors
    ) AS authors_count,
    (
        SELECT COUNT(*) FROM context_annotations
    ) AS context_annotations_count,
    (
        SELECT COUNT(*) FROM context_domains
    ) AS context_domains_count,
    (
        SELECT COUNT(*) FROM context_entities
    ) AS context_entities_count,
    (
        SELECT COUNT(*) FROM conversation_hashtags
    ) AS conversation_hashtags_count,
    (
        SELECT COUNT(*) FROM conversation_references
    ) AS conversation_references_count,
    (
        SELECT COUNT(*) FROM conversations
    ) AS conversations_count,
    (
        SELECT COUNT(*) FROM hashtags
    ) AS hashtags_count,
    (
        SELECT COUNT(*) FROM links
    ) AS links_count
```

Data Output | Messages | Notifications

	annotations_count bigint	authors_count bigint	context_annotations_count bigint	context_domains_count bigint	context_entities_count bigint	conversation_hashtags_count bigint	conversation_references_count bigint	conversations_count bigint	hashtags_count bigint	links_count bigint
1	19458972	5895176	133941462	88	29386	54613745	27801603	32347011	773865	11540704

Veľkosti tabuliek (pozor na jednotky!):

```sql
SELECT
    pg_database_size('twitter')/1024/1024/1024 AS twitter_database_GBs,
    pg_relation_size('annotations')/1024/1024 AS annotations_table_MBs,
    pg_relation_size('authors')/1024/1024 AS authors_table_MBs,
    pg_relation_size('context_annotations')/1024/1024 AS context_annotations_table_MBs,
    pg_relation_size('context_domains')/1024 AS context_domains_table_KBs,
    pg_relation_size('context_entities')/1024/1024 AS context_entities_table_MBs,
    pg_relation_size('conversation_hashtags')/1024/1024 AS conversation_hashtags_table_MBs,
    pg_relation_size('conversation_references')/1024/1024 AS conversation_references_table_MBs,
    pg_relation_size('conversations')/1024/1024 AS conversations_table_MBs,
    pg_relation_size('hashtags')/1024/1024 AS hashtags_table_MBs,
    pg_relation_size('links')/1024/1024 AS links_table_MBs
```

Data Output | Messages | Notifications

	twitter_database_gbs bigint	annotations_table_mbs bigint	authors_table_mbs bigint	context_annotations_table_mbs bigint	context_domains_table_kbs bigint	context_entities_table_mbs bigint	conversation_hashtags_table_mbs bigint	conversation_references_table_mbs bigint	conversations_table_mbs bigint	hashtags_table_mbs bigint	links_table_mbs bigint
1	29	1303	542	7694	16	3	2717	1820	7914	39	1774