

# Protokol k zadaniu 1

Valter Szűcs

Link na github: [https://github.com/valter741/pgsql\\_import](https://github.com/valter741/pgsql_import)

## 1. Opis algoritmu

Pre na-importovanie dát do databázy využívam viacero prístupov. Pri importe autorov načítavam záznamy z gzip súboru po 10 000 a následne ich pomocou copy vkladám do predpripravenej tabuľky. Táto tabuľka už má nastavené id ako primary key a keďže záznamy pridávam po 10 000 pri každom PK violation prejdem daný blok a pojednom ich popridávam do tabuľky. V prípade, že je veľa duplikátov alebo by boli rozmiestnené v každom bloku, mohlo by nám to pridávanie značne spomaliť.

Do ostatných tabuliek práve preto pridávam záznamy bez constraint-ov a následne pomocou SQL tieto constraint-y na tabuľky nastavím. Taktiež tu načítavam z gzip súboru po 10 000 a pripravím si dáta na copy. Následne všetky dáta vložím do tabuliek. Po vložení dát pomocou SQL de duplikujem tabuľky conversations, context\_domains, context\_entities a hashtags. Tým, že konverzácie nededuplikujem pred vložením v niektorých tabuľkách nám zostane zopár duplicitných záznamov ktoré nevieme jednoduchou de duplikáciou odstrániť, keďže pri importe strácame dáta, podľa ktorých by sme takéto duplikáty mohli identifikovať.

Na koniec je potrebné ešte upraviť tabuľku conversation\_hashtags, ktorá je vyplnená pomocou pomocného stĺpca tag.

Pre optimálne veľkosti tabuliek je potrebné na tabuľky context\_domains, context\_entities a hashtags spustiť vacuum full \*table\_name\*, keďže sme z nich vymazali gigantické množstvo dát. Toto však psycopg3 nevedel spustiť, tak sa to spúšťa manuálne z dbms.

## 2. Použité technológie

Pre prácu z databázou som si vybral jazyk Python, keďže som v ňom asi najzručnejší a knižnicu psycopg a práve verziu 3, keďže táto knižnica má schopnosť vykonať COPY FROM z Python tuple-ov čo značne zjednoduší a urýchly importovanie dát do databázy. Tiež som sa rozhodol použiť knižnicu json\_lines, ktorá dokáže načítať za gzip-ované súbory po riadkoch a vytvoriť z nich Python slovník.

### 3. Vysvetlenie SQL

```
"COPY annotations (conversation_id, value, type, probability) FROM STDIN")
```

Copy použitá na efektívne vkladanie viacerých riadkov do tabuliek naraz

```
"INSERT INTO authors () VALUES () ON CONFLICT DO NOTHING"
```

Insert využitý na vkladanie autorov po jednom na bloky s conflict-om

```
DELETE FROM conversations a USING (  
    SELECT MIN(ctid) as ctid, id  
    FROM conversations  
    GROUP BY id HAVING COUNT(*) > 1  
    ) b  
WHERE a.id = b.id  
AND a.ctid <> b.ctid;
```

Delete využitý na deduplikácie tabuliek. V tomto delete ako subquery si selectujeme rovnakú tabuľku ako z ktorej deletujeme, len si v subselecte group-neme riadky podľa id. Ak tu nájdeme riadky s rovnakým id ale odlišným ctid našli sme duplikát a vymažeme ho. Na deduplikovanie som našiel tento prístup ako naj efektívnejší.

```
INSERT INTO authors (id)  
SELECT DISTINCT author_id FROM conversations conv WHERE conv.author_id NOT IN (  
    SELECT id FROM authors auth WHERE auth.id = conv.author_id);
```

Insert chýbajúcich id do autorov. Jednoduchý distinct select takých author\_id z conversations, ktorý sa v authors nenachádza a ich insert.

```
UPDATE conversation_hashtags  
SET hashtag_id = hashtags.id  
FROM hashtags  
WHERE hashtags.tag = conversation_hashtags.tag;
```

Update využitý na vyplnenie hashtag\_id v conversation\_hashtags. Táto query má najdlhší priebeh a je asi aj najmenej efektívna. Jej úlohu však splní, ale môže existovať aj efektívnejšie riešenie.

Ostatné query využité v programe sú len CREATE TABLE a ALTER TABLE ADD CONSTRAINT pre vytváranie tabuliek a pridávanie constraintov

#### 4. Dĺžka trvania importu a časový opis priebehu

Dĺžka importu je v mojom prípade 90 minút a 7 sekúnd a dáta sa nachádzajú v súbore timer.csv. Na vykonanie importu bol využitý procesor AMD 5800HS (laptopová verzia) a m.2 SSD zo R/W rýchlosťou okolo 500mb/s.

Prvé približne 3 minúty 15 sekúnd prebieha importovanie dát do autorov, ďalej až do 37:30 je import dát do ostatných tabuliek okrem hashtagov keďže tie si vytváram neskôr.

Od 37:30 do 41:4 vymazávanie duplikátov z konverzácií pridávanie PK, pridávanie chovajúcich autorov a pridávanie

Do 41:41 pridáme FK pre linsk a annotations.

Do 42:59 vymažeme z conversation\_references záznamy s neplatné parent\_id a pridáme dva FK na conversations

Do 47:28 de duplikujeme context\_domains.

Do 51:17 de duplikujeme context\_entities.

Do 54:20 pridáme PK pre context\_domains a entities a 3 potrebné FK pre context\_annotations.

Do 55:25 vytvoríme tabuľku hashtags ako kópiu conversation\_hashtags, ktorá má pomocný stĺpec tag.

Do 64:23 de duplikujeme tabuľku hashtags a pridáme unique constraint.

Do 87:29 pridáme chýbajúce hashtag id do conversation\_hashtags.




Ako posledné Do 90:7 odstránime pomocný stĺpec z conversation\_hashtags a pridáme chýbajúce 2 FK tabuľky.

5. Počet a veľkosť záznamov v každej tabuľke

Počet:

	annotations bigint 	authors bigint 	context_annotations bigint 	context_domains bigint 	context_entities bigint 	conversation_hashtags bigint 	conversation_references bigint 	conversations bigint 	hashtags bigint 	links bigint 
1	19480545	5895176	134444727	88	29438	54675784	27950190	32347011	773865	11552641

Veľkosť:

	annotations text 	authors text 	context_annotations text 	context_domains text 	context_entities text 	conversation_hashtags text 	conversation_references text 	conversations text 	hashtags text 	links text 
1	1722 MB	1091 MB	10 GB	40 kB	3960 kB	8784 MB	2440 MB	8620 MB	81 MB	2024 MB