Przetwarzanie masywnych danych

Sprawozdanie

MSD – Schemat i transformacja danych

Marcin Mrugas 122580

5 kwietnia 2018

1 Schemat

Naszym zadaniem była transformacja surowych plików z danym odsłuchów i piosenek w schmat ułatwiający wykonywanie zapytań związanym z zapisanymi danymi.

Początkowo dane znajodowały się w dwóch oddzielnych tableach (Rys. 1). Końcowy schemat zawiera odsłuchy w centrum w tabeli faktów i tabele wymiarów zawierające szczegółowe dane (Rys. 2).

2 Proces transformacji

Na początku zostały stworzone table odopowiadające danym przechowywanym w plikach. Następnie dane zostały wczytane z plików za pomocą polecenia `LOAD DATA LOCAL INFILE`:

```
mysql> LOAD DATA LOCAL INFILE '/path/to/dataset/triplets_sample_20p.
    txt' INTO TABLE listen FIELDS TERMINATED BY '<SEP>' LINES
    TERMINATED BY '\n';
Query OK, 27729357 rows affected, 65535 warnings (6 min 14.93 sec)
Records: 27729357 Deleted: 0 Skipped: 0 Warnings: 27729357
```

Aby wydzielić daty do osobnej tabeli został sprawdzony zakres dat w pliku zawierającym odsłuchy. Najwczeniejsza data pochodziła ze stycznia 2001 roku, a najstarsza z grudnia 2010 roku. Tabela została wypełniona za pomocą prostego skrypu w pythonie.

```
>>> for i in range(2001,2011):
... for j in range(1,13):
... db.query('INSERT into date (year,month) values ({}, {})'.format(
    i,j))
...
```

Do tabeli odsłuchów została dodana kolumna zawierająca nowe identyfikatory daty.

W kolejnym kroku została powiązana z tableą zawierającą wszystkie lata i miesiące. Ponieważ nie został stworzony index na żadnej z kolumn trwało to dłużej niż powinno.

Następnie na tej samej zasadzie została stworzona i wypełniona danymi osoba tabela zawierająca id użytkowników. Dodano także sztuczny klucz do tabeli piosenek i stworzono index na starym identyfikatorze piosenki. Po czym dodano klucz obcy na podstawie starego identyfikatora.

Listing 1: Przykładowe zapytanie aktualizaujące klucze w tabeli faktów

```
mysql> update listen join song on listen.song_id = song.id join
    artist on song.artist_name = artist.name set listen.artist_id =
    artist.id;
Query OK, 27729357 rows affected (44 min 35.23 sec)
Rows matched: 27729357 Changed: 27729357 Warnings: 0
```

3 Czas przetwarzania i rozmiar danych

Na łączny czas przetwarzania składa się czas przetworzenia każdej z kolumn, niestety zapisanie czasu wszystkich operacji nie zostało przeprowadzone ze wzgledu na różny czas przetwarzania i zmiany w koncepcji. Dodanie lub usunięcie kolumny na tabeli odsłuchań trwało ok 6 minut. wyciągnięcie unikalnych danych z tabel utworów i odsłuchań trwało zazwyczaj poniżej minuty. Najbardziej kosztowne okazały się operacje aktualizowania obcych identyfikatorów i łączenia tabel które trwały kilkanaście minut. Tworzenie indeksów trawało kilka minut.

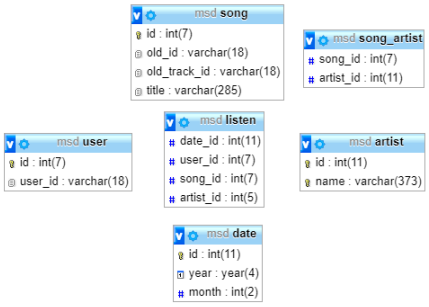
tabela	rozmiar	dane	index
artist	8 MB	3.5 MB	4.5 MB
date	16 KB	16 KB	0 KB
listen	2.6 GB	1.2 GB	1.4 GB
song	126.2 MB	94.6 MB	31.6 MB
song_artist	9.3 MB	2.9 MB	6.5 MB
user	72.2 MB	43.6 MB	28.6 MB
razem	2.9 GB	1.5 GB	1.4 GB

4 Zapytania i czas ich wykonania

1. Najpopularniejsze piosenki – popularność piosenki jako ilość odsłuchań danego utworu

   ```
   select title, count from(select song_id, count(song_id) as
       count from listen group by listen.song_id order by count(
       listen.song_id) desc) as tmp left join song on song.id =
       tmp.song_id;
   ```

Rysunek 1: Schemat przed tranformacją



Rysunek 2: Schemat po tranformacji

title	**artist**	**count**
You're The One	Dwight Yoakam	145267
Undo	Björk	129778
Revelry	Kings Of Leon	105162
Sehr kosmisch	Harmonia	84981
Horn Concerto No. 4 in E flat K495: II. Romance (Andante cantabile)	Barry Tuckwell/Academy of St Martin-in-the-Fields/Sir Neville Marriner	77632
Dog Days Are Over (Radio Edit)	Florence + The Machine	71300
Secrets	OneRepublic	58472
Canada	Five Iron Frenzy	54655
Invalid	Tub Ring	53494
Ain't Misbehavin	Sam Cooke	49073
Représente	Alliance Ethnik	48608
Catch You Baby (Steve Pitron & Max Sanna Radio Edit)	Lonnie Gordon	47345
Sincerité Et Jalousie	Alliance Ethnik	45071
Hey_ Soul Sister	Train	41712
Fireflies	Charttraxx Karaoke	39481

Tablica 1: Najpopularniejsze piosenki

user_id	count
ec6dfcf19485cb011e	1040
119b7c88d58d0c6eb0	641
b7c24f770be6b80280	637
4e73d9e058d2b1f2db	592
d7d2d888ae04d16e99	586
6d625c6557df84b60d	584
113255a012b2affeab	561
c1255748c06ee3f644	547
db6a78c78c9239aba3	529
99ac3d883681e21ea6	499
0c2932cb475b83b610	496
fef771ab021c200187	481
18c1dd917693fd929e	466
1aa4fd215aadb16096	462
b4c94d72b15d3c311c	455

count	name
201081	Coldplay
182709	Kings Of Leon
156776	Florence + The Machine
146580	Dwight Yoakam
143865	Björk
139775	The Black Keys
115877b	Jack Johnson
114233	Justin Bieber
107169	Train
106401	Alliance Ethnik
105721	OneRepublic
104996	Muse
100322	Radiohead
97448	The Killers
93881	Linkin Park

Tablica 2: Najbardziej aktywni użytkownicy oraz artyści z największą liczbą odsłuchań

Czas zapytania 27.44 sekund, zwróconych rekordów 332123.

2. Najbardziej aktywni użytkownicy - słuchacze którzy odtworzyli najwięcej unikalnych piosenek

```
select new_user_id, count(distinct new_track_id) as count from
    listen group by listen.new_user_id order by count(distinct
    listen.new_track_id) desc limit 15;
```

3. Artysta z największą liczbą odsłuchań

```
select count, name from (select artist_id, count(artist_id) as
    count from listen group by artist_id order by count(
    artist_id) desc) as tmp left join artist on tmp.artist_id =
     artist.id;
```

Czas zapytania 13.62 sekund, zwróconych rekordów: 39272.

4. Sumaryczna liczba odsłuchań w podziale na poszczególne miesiące

```
select count(*), month from listen join date on date.id =
    date_id group by date.month;
```

Czas zapytania 15.35 sekund, zwróconych rekordów: 12.

5. Wszyscy użytkownicy, którzy odsłuchali wszystkie trzy najbardziej popularne piosenki zespołu Queeni.

count	month
2353286	1
2142739	2
2351583	3
2280859	4
2357973	5
2277722	6
2353267	7
2354972	8
2281348	9
2358168	10
2278335	11
2339105	12

user_id
00832bf55ed890afeb
01cb7e60ba11f9b96e
0ac20218b5168c10b8
1084d826f98b307256
11abd6aaa54a50ed55
28daf225834bae38f8
429303f0cacab81f0c
476a5902414891326e
4cd2428f7bfcff1e24
5283f472d868bfac68
5a60f46e0d86990db2
5da28279b8926c158f
646cb63c7e2cf3f4a3
6f1b401196e9b0bd83
6ff8c66c0b69b71fba

Tablica 3: Miesięczna sumaryczna liczba odsłuchań oraz użytkownicy, którzy odsłuchali 3 piosenki zespołu Queen

```
select user.user_id from( select listen.user_id, count(distinct
     popular.song_id) as dist_count from listen join ( select
    song_id, count(song_id) as count from listen where song_id
    in ( select id from song join song_artist on song_id = song
    .id where artist_id = ( select id from artist where name =
    'Queen') ) group by song_id order by count(song_id) desc
    limit 3 ) as popular on popular.song_id = listen.song_id
    group by listen.user_id having dist_count = 3 ) as users
    join user on user.id = users.user_id
```

Czas wykonania zapytania: 4.08 sekundy, zwróconych rekordów: 34.

5 Porównanie oryginalnego i nowego schematu

Nowy schemat pozwala na efektywniejsze wykonywanie zapytań dzięki całkowitoliczbowym sztucznym identyfikatorom, które są dużo łatwiejsze w przetwarzaniu dla bazy danych niż klucze tesktowe. Dodatkowo schemat bazy jest tak rozłożony by w jednej tabeli byłaprzechowywana informacja tylko o jednej informacji, dlatego wyszukiwanie i łączenie tabel jest dużo bardziej efektywne.

6 Podsumowanie

MySQL dobrze poradził sobie z wczytaniem i przetwarzaniem danych. Jednakże w trakcie przetwarzania zdecydowałem się dodać klucze obce co dla

tej ilości danych niemiało szans na powodzenie, przez co zmarnowałem dużo czasu na oczekiwanie na wykonanie poleceń.