06.06.2021

Wydajność złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych

Julia Kwaśniak

Geoinformatyka

Rok 2, grupa 2

Przedmiot: Bazy danych

Spis treści

1. Wprowadzenie.

Celem projektu było sprawdzenie wydajności dla tabel znormalizowanych i zdenormalizowanych. Testy opierały się na bazie danych Geo z utworzoną tabelą stratygraficzną w formie małych tabel (tabela znormalizowana) oraz w formie jednej tabeli utworzonej z innych tabel (tabela zdenormalizowana). Przeprowadzone zostały testy wydajnościowe bez indeksów, a później z indeksami, a następnie dane z owch testów zebrane zostały w formie tabeli.

2. Tabela stratygraficzna.

Tabela stratygraficzna składa się z pięciu oddzielnych tabel: GeoEon, GeoEra, GeoOkres, GeoEpoka, GeoPiętro. Każda z nich posiada klucz główny oraz klucz obcy, który nie może być wartością NULL, z uwagi na późniejsze połączenie powyższych tabel.

```
--1.Utworzenie tabel

create table GeoEon(id_eon integer primary key, nazwa_eon varchar (15));
create table GeoEra(id_era integer primary key, id_eon integer not null, nazwa_era varchar(15));
create table GeoOkres(id_okres integer primary key, id_era integer not null, nazwa_okres varchar(30));
create table GeoEpoka(id_epoka integer primary key, id_okres integer not null, nazwa_epoka varchar(15));
create table GeoPietro(id_pietro integer primary key, id_epoka integer not null, nazwa_pietro varchar(30));


--2.Dodanie kluczy obcych

alter table GeoEra
add foreign key (id_eon) references GeoEon(id_eon);
alter table GeoOkres
add foreign key (id_era) references GeoEra(id_era);
alter table GeoEpoka
add foreign key (id_okres) references GeoOkres(id_okres);
alter table GeoPietro
add foreign key (id_epoka) references GeoEpoka(id_epoka);
```

Następnie do tabel zostały dodane wartości.

```
--3.Dodanie wartości do tabel

insert into GeoEon values(1, 'Farenozoik');

insert into GeoEra values(1, 1, 'Kenozoik');
insert into GeoEra values(2, 1, 'Mezozoik');
insert into GeoEra values(3, 1, 'Paleozoik');

insert into GeoOkres values(1, 1, 'Czwartorząd');
insert into GeoOkres values(2, 1, 'Trzeciorząd(Neogen)');
insert into GeoOkres values(3, 1, 'Trzeciorząd(Paleogen)');
insert into GeoOkres values(4, 2, 'Kreda');
insert into GeoOkres values(5, 2, 'Jura');
insert into GeoOkres values(6, 2, 'Trias');
insert into GeoOkres values(7, 3, 'Perm');
insert into GeoOkres values(8, 3, 'Karbon');
insert into GeoOkres values(9, 3, 'Dewon');

insert into GeoEpoka values(1, 1, 'Halocen');
insert into GeoEpoka values(2, 1, 'Plejstocen');
insert into GeoEpoka values(3, 2, 'Pliocen');
insert into GeoEpoka values(4, 2, 'Miocen');
insert into GeoEpoka values(5, 3, 'Oligocen');
insert into GeoEpoka values(6, 3, 'Eocen');
insert into GeoEpoka values(7, 3, 'Paleocen');
insert into GeoEpoka values(8, 4, 'Górna');
insert into GeoEpoka values(9, 4, 'Dolna');
insert into GeoEpoka values(10, 5, 'Górna');
insert into GeoEpoka values(11, 5, 'Środkowa');
insert into GeoEpoka values(12, 5, 'Dolna');
insert into GeoEpoka values(13, 6, 'Górna');
insert into GeoEpoka values(14, 6, 'Środkowa');
insert into GeoEpoka values(15, 6, 'Dolna');
insert into GeoEpoka values(16, 7, 'Loping');
```

Z powyższych tabel utworzona została tabela w postaci zdenormalizowanej o nazwie GeoTabela.

```
CREATE TABLE GeoTabela AS (SELECT * FROM GeoPietro
                           NATURAL JOIN GeoEpoka
                           NATURAL JOIN GeoOkres
                           NATURAL JOIN GeoEra
                           NATURAL JOIN GeoEon );
```

3. Testy wydajności.

Testy wykonane zostały w programie PostgreSQL

Utworzone zostały tabele: Dziesiec zawierająca liczby od 0 do 9, aby na jej podstawie mogła zostać utworzona tabela Milion zawierająca milion rekordów.

```
create table Dziesiec(cyfra int, bit int);
insert into Dziesiec(cyfra) values (0), (1), (2), (3), (4), (5), (6), (7), (8), (9);

CREATE TABLE Milion(liczba int,cyfra int, bit int);
INSERT INTO Milion
SELECT a1.cyfra +10* a2.cyfra +100*a3.cyfra + 1000*a4.cyfra + 10000*a5.cyfra + 10000*a6.cyfra
AS liczba , a1.cyfra AS cyfra, a1.bit AS bit
FROM Dziesiec a1, Dziesiec a2, Dziesiec a3, Dziesiec a4, Dziesiec a5, Dziesiec a6 ;
```

Parametry komputera i programu PostgreSQL:

CPU: AMD A12-9720P RADEON R7, 12 COMPUTE CORES 4C+8G 2.70 GHz

RAM: 8GB

System operacyjny: Windows 10

PostgreSQL: Wersja 13.3-2

Kryteria testów:

1. Zapytanie 1 (1 ZL), którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej.

   ```
   explain SELECT COUNT(*) FROM Milion INNER JOIN GeoTabela ON
   (mod(Milion.liczba,77)=(GeoTabela.id_pietro));
   ```

2. Zapytanie 2 (2 ZL), którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, reprezentowaną przez złączenia pięciu tabel.

   ```
   explain SELECT COUNT(*) FROM Milion INNER JOIN GeoPietro ON
   (mod(Milion.liczba,68)=GeoPietro.id_pietro) NATURAL JOIN GeoEpoka NATURAL JOIN
   GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon;
   ```

3. Zapytanie 3 (3 ZG), którego celem jest złączenie syntetycznej tablicy miliona wyników , przy czym złączenie jest wykonywane poprzez zagnieżdżenie skorelowane.
   ```
   explain SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,77)=
   (SELECT id_pietro FROM GeoTabela WHERE mod(Milion.liczba,77)=(id_pietro));
   ```

4. Zapytanie 4 (4 ZG), którego celem jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej.

```
explain SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,77)=
(SELECT GeoPietro.id_pietro FROM GeoPietro NATURAL JOIN GeoEpoka NATURAL JOIN
GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon);
```

4. Wyniki testów.

Testy zostały przeprowadzone na każdym zapytaniu 10 razy oraz kolejne 10 razy na zapytaniach z indeksami.

Tabela 1.

1zl		z indexem		2zl		z indexem		3zl		z indexem		4zl		z indexem
1	441	549		1	927	954		1	35587	42959		1	506	611
2	424	722		2	1001	978		2	38073	34119		2	730	721
3	635	655		3	1017	975		3	34332	35462		3	611	642
4	814	690		4	999	1069		4	35385	35316		4	631	709
5	818	664		5	897	1066		5	35291	37473		5	838	650
6	730	660		6	1109	990		6	35248	35471		6	714	510
7	829	658		7	944	783		7	35566	37533		7	671	532
8	696	636		8	821	1094		8	35407	35630		8	673	533
9	692	719		9	961	2765		9	35610	34878		9	650	691
10	684	628		10	965	1170		10	34369	33986		10	688	582
min:	424	549		min:	821	783		min:	34332	33986		min:	506	510
średnia:	676,3	658,1		średnia:	964,1	1184,4		średnia:	35486,8	36282,7		średnia:	671,2	618,1

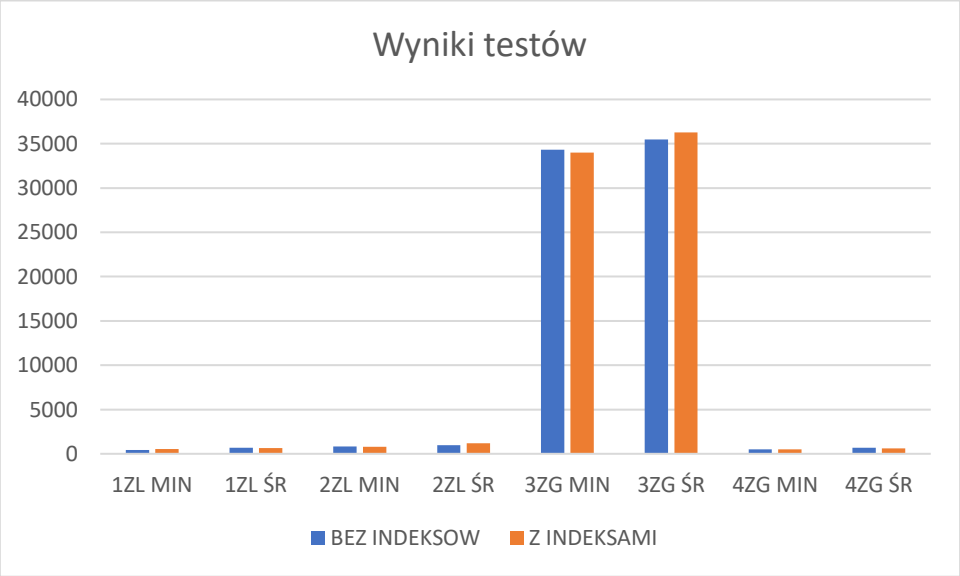
Wyniki pojedynczych testów dla każdego zapytania

Tabela 2.

	1ZL		2ZL		3ZG		4ZG	
	MIN	**ŚR**	**MIN**	**ŚR**	**MIN**	**ŚR**	**MIN**	**ŚR**
BEZ INDEKSOW	424	676,3	821	964,1	34332	35486,8	506	671,2
Z INDEKSAMI	549	658,1	783	1184,4	33986	36282,7	510	618,1

Wyniki minimalne i średnie wyliczone z testów z Tabeli 1.

Wykres 1.



Wykres wyników z Tabeli 2.

5. Wnioski.

Po dodaniu indeksów wydajność zapytań 2 i 3 lekko się zmniejszyła, a zapytań 1 i 4 lekko zwiększyła. Postać znormalizowana jest szybsza niż zdenormalizowana. Można zauważyć, że w przypadku zapytania trzeciego, gdzie złączenie jest wykonywane przez zagnieżdżenie skorelowane a czas wykonania był rzędu 34000-35000 ms, wydajność jest gorsza niż w pozostałych zapytaniach, gdzie czas wykonania nie przekraczał zazwyczaj 2000ms.