|  |
| --- |
|  |
| ĆWICZENIE 9 |
|  |

# WSTĘP

Cel ćwiczenia obejmuje sprawdzenie wydajności dla złączeń i zagnieżdżeń skolerowanych dla systemów zarządzania bazami danych PostgreSQL i SQL Server.

Testy wydajnościowe zostały przeprowadzone na przykładzie wymiaru czasu geologicznego/jednostek geochronologicznych – podstawę konstrukcji baz danych geologicznych (zgodnych z ideą konstrukcji dużych baz danych oraz hurtowni w postaciach znormalizowanej i zdenormalizowanej).

# SPECYFIKACJA

CPU: Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz

RAM: 16GB

SSD: ADATA SX600LNP

System operacyjny: Windows 10 Home

Systemy zarządzania bazami:

* PostgreSQL-15.3
* Microsoft SQL Server 19.0.2

# OPIS

Konstrukcyjnie baza danych składała się z tabeli geochronologicznej (jednostki gechronologiczne z wymiarem czasowym). Wymiar czasowy to w podanej bazie:

* GeoEon
* GeoEra
* GeoOkres
* GeoEpoka
* GeoPietro

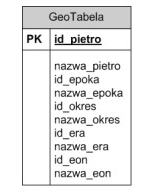
Tabela może występować w dwóch typach:

- schemacie znormalizowanym:



Rysunek 1 Schemat znormalizowany (Jajeśnica Ł., Piórkowoski A.)

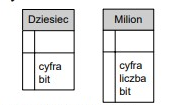
- schemacie zdenormalizowanym:



Rysunek 2 Schemat zdenormalizowany (Jajeśnica Ł., Piórkowoski A..)

Rysunek 2 przedstawia *GeoTabelę* łączącą wszystkie dane z poszczególnych tabel, zapewnia ona szybką odpowiedź na zapytania. Tabela została uzyskana przez złączenie naturalne.

Testy wymagały stworzenia dodatkowo tabeli Dziesiec i Milion z syntetycznymi danymi o rozkładzie jednostajnym wypełnionymi liczbami naturalnymi od 0 do 999 999 (rysunek 3).



Rysunek 3 Schemat tabel pomocniczych (Jajeśnica Ł., Piórkowoski A.)

Procedura testów skupiła się na sprawdzeniu wpływu normalizacji na zapytania złożone. Stąd proces podzielono na 2 etapy: pierwszy bez nałożonych indeksów i drugi z nałożonymi indeksami na wszystkie kolumny w złączeniu. Każdy aspekt był sprawdzany 10 razy.

## Zapytania

1. złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym do warunku złączenia dodano operację modulo, dopasowującą zakresy wartości złączanych kolumn:

SELECT COUNT(\*) FROM Milion INNER JOIN GeoTabela ON

(mod(Milion.liczba,77)=(GeoTabela.id\_pietro));

--

set statistics time on;

select COUNT(\*) from geochronologia.Milion INNER JOIN geochronologia.GeoTabela on Milion.liczba%77 = GeoTabela.id\_pietro;

set statistics time off;

1. złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, reprezentowaną przez złączenia pięciu tabel:

SELECT COUNT(\*) FROM Milion INNER JOIN GeoPietro ON

(mod(Milion.liczba,77)=GeoPietro.id\_pietro) NATURAL JOIN GeoEpoka NATURAL JOIN

GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon;

--

set statistics time on;

select COUNT(\*) from geochronologia.Milion inner join geochronologia.GeoPietro on

(geochronologia.Milion.liczba%77=GeoPietro.id\_pietro)

inner join geochronologia.GeoEpoka on GeoPietro.id\_epoka=GeoEpoka.id\_epoka

inner join geochronologia.GeoOkres on GeoEpoka.id\_okres= GeoOkres.id\_okres

inner join geochronologia.GeoEra on GeoEra.id\_era=GeoOkres.id\_era

inner join geochronologia.GeoEon on GeoEon.id\_eon=GeoEra.id\_eon

set statistics time off;

1. złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym złączenie jest wykonywane poprzez zagnieżdżenie skorelowane:

SELECT COUNT(\*) FROM Milion WHERE mod(Milion.liczba,77)=

(SELECT id\_pietro FROM GeoTabela WHERE mod(Milion.liczba,77)=(id\_pietro));

--

set statistics time on;

select COUNT(\*) from geochronologia.Milion where Milion.liczba%77 =

(select id\_pietro from geochronologia.GeoTabela where Milion.liczba%77=id\_pietro)

set statistics time off;

1. złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, przy czym złączenie jest wykonywane poprzez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem tabel poszczególnych jednostek geochronologicznych:

SELECT COUNT(\*) FROM Milion WHERE mod(Milion.liczba,77) in

(SELECT GeoPietro.id\_pietro FROM GeoPietro NATURAL JOIN GeoEpoka NATURAL JOIN GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon);

set statistics time on;

select COUNT(\*) from geochronologia.Milion where Milion.liczba%77 in

(select GeoPietro.id\_pietro from geochronologia.GeoPietro

inner join geochronologia.GeoEpoka on GeoPietro.id\_epoka = GeoEpoka.id\_epoka

inner join geochronologia.GeoOkres on GeoEpoka.id\_okres = GeoOkres.id\_okres

inner join geochronologia.GeoEra on GeoEra.id\_era = GeoOkres.id\_era

inner join geochronologia.GeoEon on GeoEon.id\_eon = GeoEra.id\_eon)

set statistics time off;

\*PostgreSQL

\*SQL Server

# WYNIKI TESTÓW

Zestawienie czasów minimalnych I średnich zostało przedstawione w tabeli 1:

\*1 ZL i 3 ZG to pytania w postaci zdenormalizowanej (3 ZG zagnieżdżenie skolerowane)

\*2 ZL i 4 ZG to pytania w postaci znormalizowanej (4 ZG zagnieżdżenie skolerowane)

Tabela 1 Wyniki testów [ms]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 ZL | | 2 ZL | | 3 ZG | | 4 ZG | |
|  | MIN | AVG | MIN | AVG | MIN | AVG | MIN | AVG |
| BEZ INDEKSÓW | | | | | | | | |
| SQL SERVER | 126 | 182 | 140 | 186 | 172 | 203 | 143 | 215 |
| PostgreSQL | 166 | 200 | 288 | 331 | 9128 | 9206 | 151 | 173 |
| Z INDEKSAMI | | | | | | | | |
| SQL SERVER | 47 | 105 | 77 | 154 | 48 | 110 | 32 | 117 |
| PostgreSQL | 144 | 188 | 222 | 259 | 9201 | 9888 | 161 | 187 |

Aby lepiej zobrazować różnice we wpływie indeksowania, stworzono wykres 1:

Wykres 1 Wizualizacja wyników testu (opracowanie własne)

# WNIOSKI

Wyniki testów przeprowadzonych w SQL Serverze wykazują, że indeksacja przyspieszyła proces zapytań. Szczególną różnicę o ponad 100ms (przy średniej) widać w zapytaniu 3 (zdenormalizowane, zagnieżdżenie skolerowane) i zapytaniu 4 (znormalizowane; zagnieżdżenie skolerowane).

Wyniki testów przeprowadzonych w PostgreSQL ukazują, że dla pierwszych dwóch zapytań (odpowiednio w postaci zdenormalizowanej i znormalizowanej) indeksacja przyspieszyła proces wykonywania zapytania. Natomiast przy zapytaniach 3 i 4 opartych na zagnieżdżeniu skolerowanym czas wykonywania zapytań w kolumach indeksowanych był dłuższy niż w kolumnach nieindeksowanych. Różnica ta nie jest znaczna w przypadku zapytania 4.

Wyniki ukazują również, że SQL Server jest ogólnie szybszy od PostgreSQL. Szczególnie odznacza się to w zapytaniu 3. Jedyne odstępstwo to zagnieżdżenie skolerowane przy PostgreSQL, gdzie następuje wydłużenie czasu dla optymalizacji.

Postać zdenormalizowana jest generalnie wydajniejsza od znormalizowanej (biorąc pod uwagę średnie).

Dodatkowe spostrzeżeniem jest fakt, że zagnieżdżenia skolerowane (szczególnie 3 ZG w postaci zdenormalizowanej) są wolniejsze od złączeń

# BIBLIOGRAFIA

Jajeśnica Ł., Piórkowoski A., WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH, Akademia Górniczo – Hutnicza, Katedra Geoinformatyki i Informatyki Stosowanej; Studia Informatica Vol. 31, No. 2A, Kraków 2010, s. 445÷456.