15.06.2022

----------Bazy danych----------

Zuzanna Kiełbasa

1.Wprowadzenie

Sprawozdanie wykonane zostało na podstawie artykułu Mgr.inż Łukasza Jaješnica oraz *dr hab. Inż* Adama Piórkowskiego.

Ma ono na celu przybliżenie aspektów wydajnościowych złączeń, zapytań oraz zagnieżdżeń danych zindekowanych oraz niezindeksowanych w dwóch różnych środowiskach. Badania przeprowadzone zostały na podstawie tabeli stratygraficznej(wymiaru czasu geochronologicznego). Poddane analizie zostaną również dwa

systemy zarządzania bazami danych dzięki czemu będziemy mieć pogląd na różnice w wydajności baz zagwarantowanych przez danych producentów.

2.Konfiguracja sprzętowa i programowa

Procesor (CPU) : AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz

Zainstalowana pamięć (RAM) : 16,0 GB

Typ systemu: 64-bitowy system operacyjny, procesor x64

System Operacyjny: Windows 10

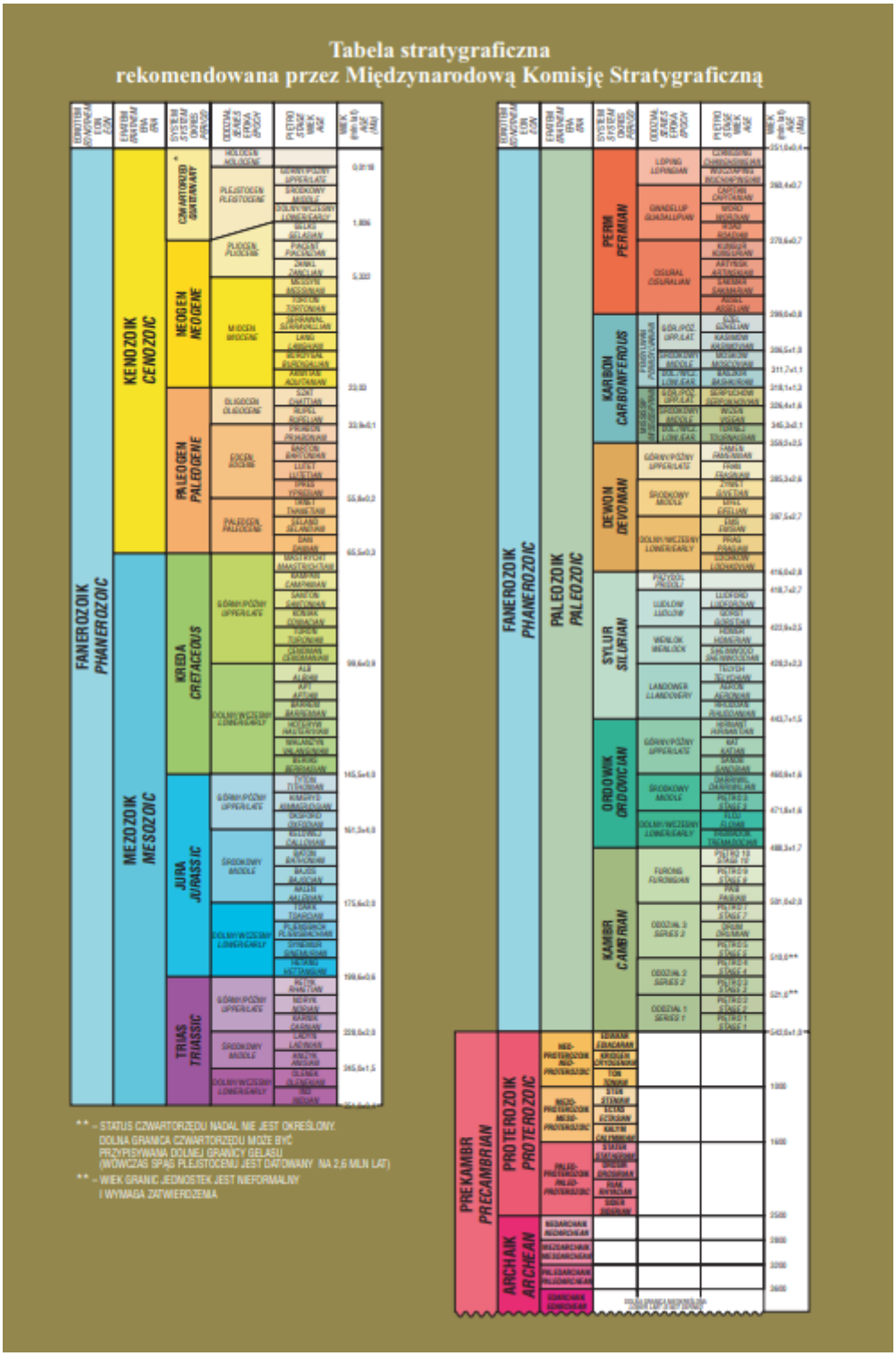
Wykorzystane systemy zarządzania bazami danych:

PostgreSQL 14

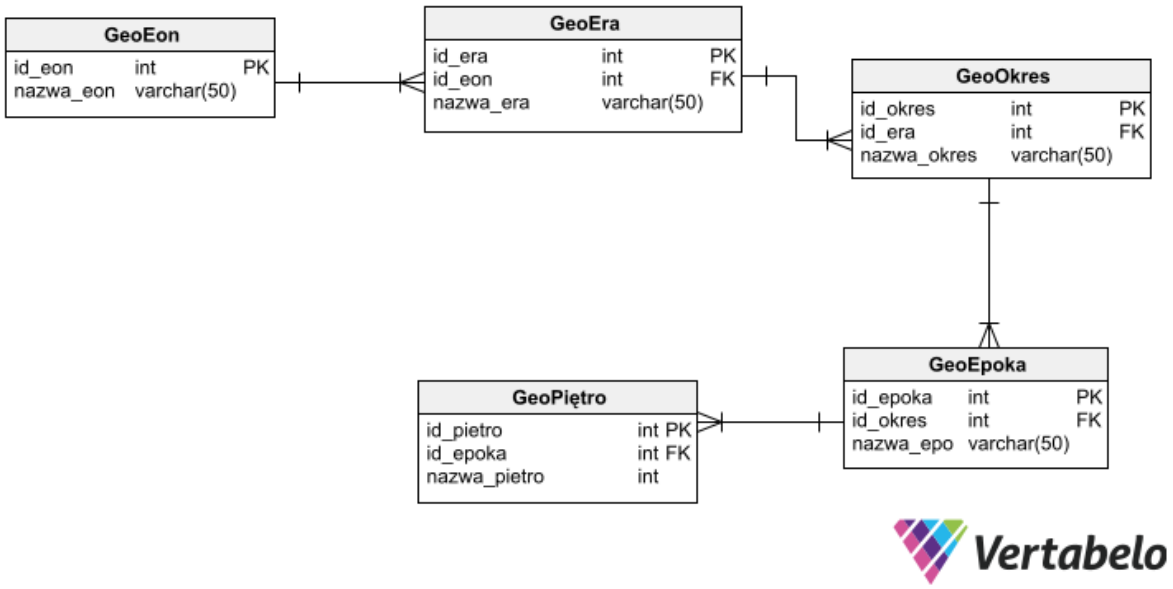
SQL Server Management Studio 18

3.Dane

W testach wykorzystaliśmy poniższą tabele stratygraficzną która obrazuje przebieg historii Ziemi na podstawie następstw procesów geologicznych i układów warstw skalnych. Dzieli się ona na eony, ery, okresy, epoki oraz piętra.



Rys.0 Tabela stratygraficzna

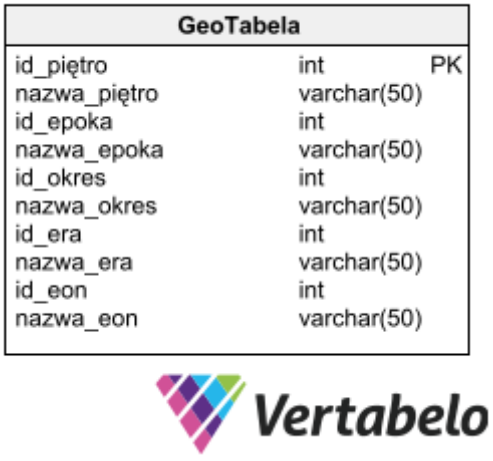


Rys.1 GeoTabela – postać znormalizowana

Powyżej przedstawiony został schemat znormalizowanej (*tzw. płatka śniegu*) tabeli stratygraficznej.

Na podstawie wyżej przedstawionych tabel powstała zawarta poniżej zdenormalizowana postać w schemacie gwiazdy. Złącze naturalne umożliwiło automatyczne wypełnienie tabeli z rys.2 danymi z tabel powyższych.

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



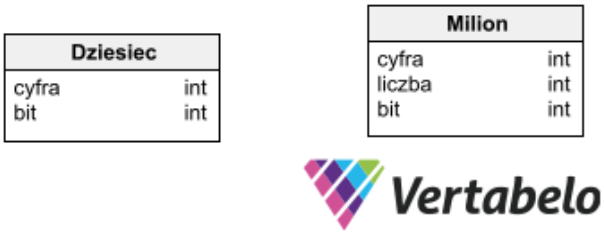
Rys.2 Geotabela – postać zdenormalizowana

Dzięki stworzeniu jednej GeoTabeli mamy szybszy oraz łatwiejszy dostęp do wszelkich danych poprzez proste zapytania.

Na potrzeby testów wydajnościowych utworzona została tabela Milion oraz Dziesięć, obie wypełnione kolejnymi liczbami naturalnymi. Tabela Milion stworzona została na podstawie auto złączenia z tabelą Dziesięć . Tabela Milion wypełniona jest liczbami od 0 do 999 990 a tabela Dziesięć w zakresie 0 do 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 ;
```

```
CREATE TABLE Dziesiec
(cyfra int,
bit int);
SELECT * FROM Dziesiec
INSERT INTO Dziesiec VALUES(0,00000);
INSERT INTO Dziesiec VALUES(1,00001);
INSERT INTO Dziesiec VALUES(2,00010);
INSERT INTO Dziesiec VALUES(3,00011);
INSERT INTO Dziesiec VALUES(4,00100);
INSERT INTO Dziesiec VALUES(5,00101);
INSERT INTO Dziesiec VALUES(6,00110);
INSERT INTO Dziesiec VALUES(7,00111);
INSERT INTO Dziesiec VALUES(8,01000);
INSERT INTO Dziesiec VALUES(9,01001);
```



Rys.3 Schemat tabel Dziesiec i Milion

4. Kryterium testów wydajności.

W teście wykonaliśmy szereg zapytań które miały na celu sprawdzenie wydajności złączeń, zapytań oraz zagnieżdżeń zindekoswanych i niezindeksowanych tabelą stratygraficzną w wersji zdenormalizowanej i znormalizowanej. Wprowadzono dwa etapy analizy, pierwszy w formie niezindeksowanej oraz drugi w formie zindeksowanej.

Poniżej przedstawiono cztery zapytania które miały na celu zbadanie wpływu normalizacji a także indeksowania na podane zapytania.

→ Zapytanie 1 (ZL).

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

```
--ZL 1
SELECT COUNT(*) FROM Milion INNER JOIN GeoTabela ON
(mod(Milion.liczba,68)=(GeoTabela.id_pietro));
```

→ Zapytanie 2 (ZL).

Celem zapytania 2 jest 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,68)=GeoPietro.id_pietro) NATURAL JOIN GeoEpoka NATURAL JOIN
GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon;
```

→ Zapytanie 3 (ZL).

Ma na celu 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,68)=
(SELECT id_pietro FROM GeoTabela WHERE mod(Milion.liczba,68)=(id_pietro));
```

→ Zapytanie 4 (ZL).

Celem tego zapytania jest złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, przy czym złączenie jest wykonywane poprzez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem ta bel poszczególnych jednostek geochronologicznych:

```
SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,68)IN
(SELECT GeoPietro.id_pietro FROM GeoPietro NATURAL JOIN GeoEpoka NATURAL JOIN
GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon);
```

5. Wyniki testów

Każdy test został wykony wielokrotnie:

SQL Server - Bez Indeksu				
Pomiary	1 ZL	2 ZL	3 ZG	4 ZG
1	128	151	3834	59
2	131	141	3320	120
3	144	164	3719	104
4	111	131	3662	169
5	89	103	3684	78
6	102	132	3574	136
7	111	158	3572	132
8	99	160	3601	155
MIN	89	103	3320	59
AVG	114	143	3621	119

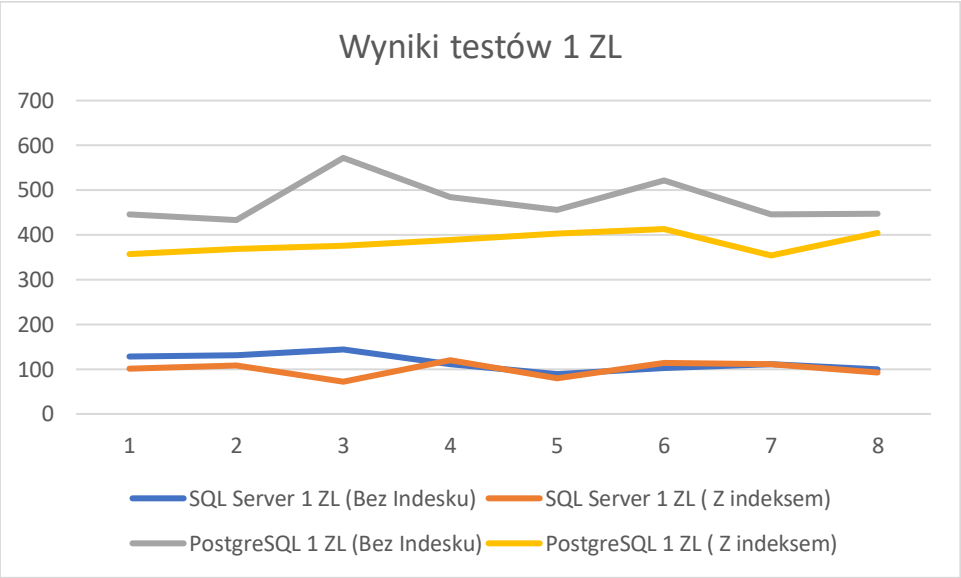
SQL Server - Z Indeksem				
Pomiary	1 ZL	2 ZL	3 ZG	4 ZG
1	101	128	3274	92
2	108	84	2980	88
3	72	134	3474	63
4	120	114	3294	124
5	79	106	3618	132
6	114	119	3512	79
7	111	131	3412	105
8	93	93	3023	116
MIN	72	84	2980	63
AVG	100	114	3323	100

Jednostka pomiarowa to ms.

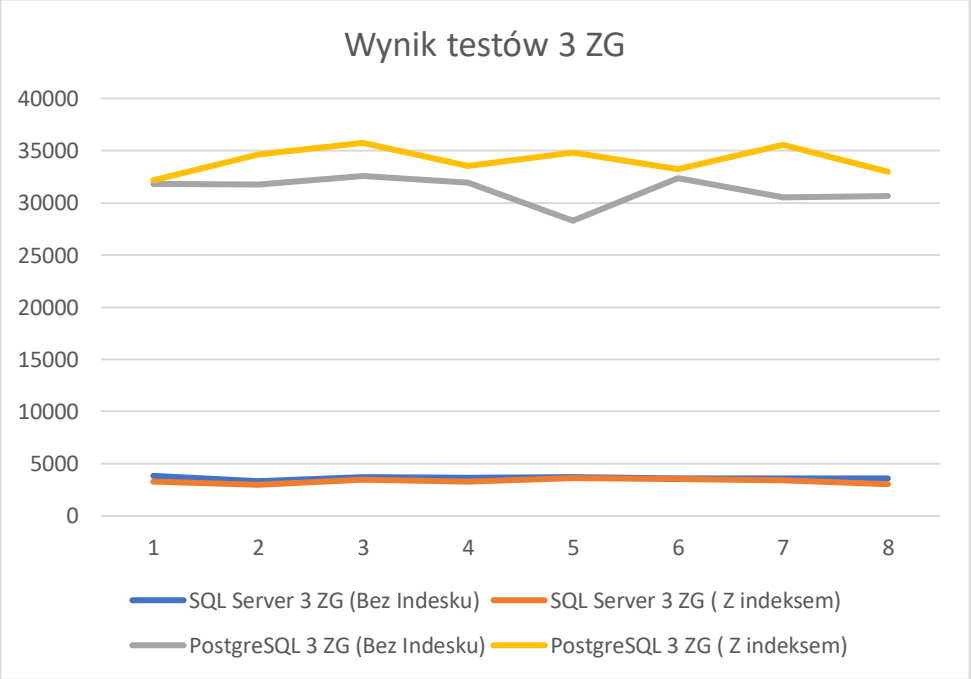
PostgreSQL - Bez Indeksu				
Pomiary	1 ZL	2 ZL	3 ZG	4 ZG
1	446	1008	31790	428
2	433	1108	31729	455
3	572	920	32569	443
4	484	1104	31924	566
5	456	934	28281	408
6	521	942	32345	492
7	446	913	30543	430
8	447	994	30623	475
MIN	433	913	28281	428
AVG	476	990	31226	462

PostgreSQL - Z Indeksem				
Pomiary	1 ZL	2 ZL	3 ZG	4 ZG
1	357	947	32158	406
2	369	829	34648	435
3	375	1002	35745	418
4	389	893	33552	423
5	403	926	34827	469
6	413	879	33214	453
7	354	834	35561	413
8	405	931	32981	498
MIN	354	829	32158	406
AVG	383	905	34086	439

Jednostka pomiarowa to ms.



Wykres nr.1



Wykres nr.2

6.Wnioski

Szereg przeprowadzonych testów pozwala nam wyciągnąć następujące wnioski:

Postać zdenormalizowana okazała się wydajniejsza w większości wypadków, wyjątek stanowi zadanie 3ZG w środowisku PostgreSQL gdzie czas obliczeń jest znacznie wydłużony.

Praktycznie w każdym zapytaniu skorzystanie z indeksów uprawniło prędkość wykonywania operacji z zapytaniami zagnieżdżonymi a także złączeniowymi z wyjątkiem zapytania zagnieżdżonego zdenormalizowanego 3 ZG w którym średnia wyników czasowych z indeksami przewyższa o kilkanaście procent średnią postać bez indeksowania. Występuje spadek wydajności procesu obliczeniowego o około 9%.

Biorąc pod uwagę dane które zostały użyte do naszych testów możemy stwierdzić który system związany z zarządzaniem bazami danych jest najwydajniejszy.

Analizując wykres nr.1 zauważamy różnice między prędkością wykonywania zapytań znormalizowanych 1ZL gdzie przoduje SQL Server bez wykorzystania indeksów a także z nimi. Wykres nr.2 przedstawia również kolosalną różnice między PostgreSQL a SQL Server.

7.Bibliografia

Mgr.inż.Łukasz JAJEŚNICA, dr hab. Inż. Adam PIÓRKOWSKI - WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH

Dr Inż. Michał Lupa – Bazy Danych 2022

Aktualna tabela stratygraficzna na stronie International Comission on Stratigraphy [dostęp 2021-06-15].