Wydajność złączeń oraz zagnieżdżeń dla schematów znormalizowanych oraz zdenormalizowanych

Zuzanna Szurman, nr indeksu: 416649

1. Wprowadzenie:

W sprawozdaniu przedstawiono analizę oraz porównanie efektywności wykonań zapytań SQL. Test przeprowadzono na schemacie znormalizowanym oraz zdenormalizowanym dla złączeń oraz zagnieżdżeń skorelowanych. Schemat znormalizowany to tabela stratygraficzna przedstawiona zgodnie z zasadami trzeciej postaci normalnej (3NF), natomiast schemat zdenormalizowany zawiera te same informacje, ale nie spełnia wymogów normalizacji.

Wykonano cztery zapytania dla każdej wersji bazy danych: SQL Server oraz PostgreSQL. Dwa z tych zapytań to złączenia, przy czym jedno zostało wykonane na schemacie znormalizowanym, a drugie na zdenormalizowanym. Pozostałe dwa zapytania to zagnieżdżenia skorelowane, adekwatnie wykonane dla dwóch różnych schematów.

Dodatkowo, zbadano wpływ indeksów klastrowanych i nieklastrowanych na wydajność operacji złączeń i zagnieżdżeń.

Wykonywane testy mają na celu odpowiedź na następujące pytania:
1. Czy schemat znormalizowany czy zdenormalizowany jest szybszy?
2. Czy indeksowanie zmienia wydajność złączeń bądź zagniedźeń?
3. Czy rodzaj bazy danych wpływa na wydajność wykonywanych zapytań?

2. Konfiguracja sprzętowa i programowa

CPU: 13th Gen Intel(R) Core(TM) i9-13900H 2.60 GHz

RAM: 32 GB

S.O.: Windows 11

SQLServer 2022 (RTM) - 16.0.1000.6 (X64)

PostgreSQL 16.3

3. Kryteria testów

Cały kod do sprawozdania umieszczono w plikach, osobno dla SQLServer oraz PostgresSQL.

Dla czterech zapytań wykonano trzy odrębne testy badające wpływ indeksów na wydajność złączeń i zagnieżdżeń. Były nimi:
- indeksy nieklastrowane będące kluczami głównymi tabeli geochronologicznych
- indeksy nieklastrowane dla tabel z pierwszego testu oraz indeks nieklastrowany nałożony na tabele Milion
- indeksy klastrowane nałożone na wszystkie tabele

Jako, że PostgresSQL formalnie nie obsługuje indeksów klastrowanych, użyto komendy „cluster”, która naśladuje zachowanie indeksu klastrowanego, po wcześniejszym utworzeniu indeksu nieklastrowanego.

Zapytanie 1: złączenie syntetycznej tabeli Milion z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym do warunku złączenia dodano operację modulo, dopasowującą zakresy wartości złączonych kolumn.

SQLserver:

```
set statistics time on
    SELECT COUNT(*) FROM Milion
    INNER JOIN tabela.GeoTabela ON Milion.liczba%68 = GeoTabela.id_pietro;
set statistics time off
```

PostgresSQL:

```
EXPLAIN ANALYZE
SELECT COUNT(*)
FROM Milion
INNER JOIN GeoTabela ON Milion.liczba % 68 = GeoTabela.id_pietro;
```

Zapytanie 2: złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej, reprezentowaną przez złączenia pięciu tabel

SQLServer:

```
set statistics time on
SELECT COUNT(*)
FROM Milion
INNER JOIN tabela.GeoPietro ON Milion.liczba % 68 = GeoPietro.id_pietro
INNER JOIN tabela.GeoEpoka ON GeoPietro.id_epoka = GeoEpoka.id_epoka
INNER JOIN tabela.GeoOkres ON GeoEpoka.id_okres = GeoOkres.id_okres
INNER JOIN tabela.GeoEra ON GeoOkres.id_era = GeoEra.id_era
INNER JOIN tabela.GeoEon ON GeoEra.id_eon = GeoEon.id_eon;
set statistics time off
```

PostgresSQL:

```
EXPLAIN ANALYZE
SELECT COUNT(*)
FROM Milion
INNER JOIN GeoPietro ON Milion.liczba % 68 = GeoPietro.id_pietro
INNER JOIN GeoEpoka ON GeoPietro.id_epoka = GeoEpoka.id_epoka
INNER JOIN GeoOkres ON GeoEpoka.id_okres = GeoOkres.id_okres
INNER JOIN GeoEra ON GeoOkres.id_era = GeoEra.id_era
INNER JOIN GeoEon ON GeoEra.id_eon = GeoEon.id_eon;
```

Zapytanie 3: złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej, przy czym złączenie jest wykonywane poprzez zagnieżdżenie skorelowane

SQLServer:

```
set statistics time on
SELECT COUNT(*)
FROM Milion
WHERE Milion.liczba % 68 = (SELECT id_pietro
                            FROM tabela.GeoTabela
                            WHERE Milion.liczba % 68 = id_pietro);
set statistics time off
```

PostgresSQL:

```
EXPLAIN ANALYZE
SELECT COUNT(*)
FROM Milion
WHERE Milion.liczba % 68 = (
    SELECT id_pietro
    FROM GeoTabela
    WHERE Milion.liczba % 68 = id_pietro
);
```

Zapytanie 4: 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:

SQLServer:

```
set statistics time on
SELECT COUNT(*)
FROM Milion
WHERE Milion.liczba % 68 = (
    SELECT GeoPietro.id_pietro
    FROM tabela.GeoPietro
    INNER JOIN tabela.GeoEpoka ON GeoPietro.id_epoka = GeoEpoka.id_epoka
    INNER JOIN tabela.GeoOkres ON GeoEpoka.id_okres = GeoOkres.id_okres
    INNER JOIN tabela.GeoEra ON GeoOkres.id_era = GeoEra.id_era
    INNER JOIN tabela.GeoEon ON GeoEra.id_eon = GeoEon.id_eon
    WHERE Milion.liczba % 68 = GeoPietro.id_pietro
);
set statistics time off
```

PostgresSQL:

```
EXPLAIN ANALYZE
SELECT COUNT(*)
FROM Milion
WHERE Milion.liczba % 68 = (
    SELECT GeoPietro.id_pietro
    FROM GeoPietro
    INNER JOIN GeoEpoka ON GeoPietro.id_epoka = GeoEpoka.id_epoka
    INNER JOIN GeoOkres ON GeoEpoka.id_okres = GeoOkres.id_okres
    INNER JOIN GeoEra ON GeoOkres.id_era = GeoEra.id_era
    INNER JOIN GeoEon ON GeoEra.id_eon = GeoEon.id_eon
    WHERE Milion.liczba % 68 = GeoPietro.id_pietro
);
```

4. Wyniki

Każde zapytanie zostało wykonane 5 razy dla każdego rodzaju indeksów oraz bazy danych. Wyniki tych zapytań przedstawia poniższa tabela.

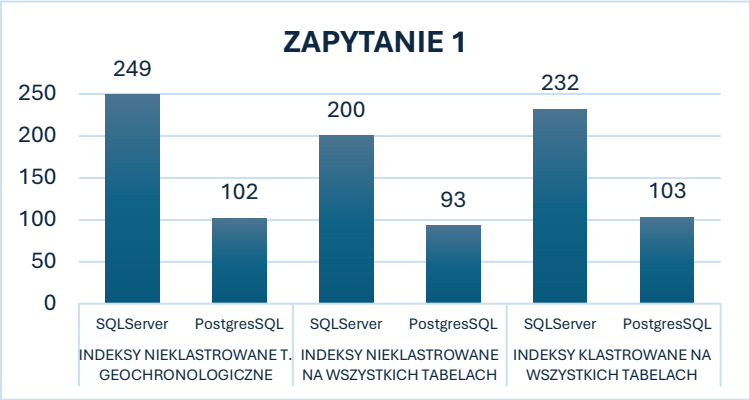
		ZAPYTANIE 1		ZAPYTANIE 2		ZAPYTANIE 3		ZAPYTANIE 4	
		MIN [ms]	ŚR [ms]	MIN	ŚR	MIN	ŚR	MIN	ŚR
INDEKSY NIEKLASTROWANE NA T. GEOCHRONOLOGICZNYCH	SQLServer	219	**249**	172	**254**	187	**215**	16252	**20197**
	PostgresSQL	98	**102**	197	**211**	24776	**25122**	37009	**38896**
INDEKSY NIEKLASTROWANE NA WSZYSTKICH TABELACH	SQLServer	186	**200**	187	**225**	187	**203**	13202	**18012**
	PostgresSQL	82	**93**	210	**229**	20109	**23687**	37465	**38915**
INDEKSY KLASTROWANE NA WSZYSTKICH TABELACH	SQLServer	203	**232**	188	**207**	155	**203**	34812	**38963**
	PostgresSQL	96	**103**	487	**528**	16570	**16996**	36746	**37247**

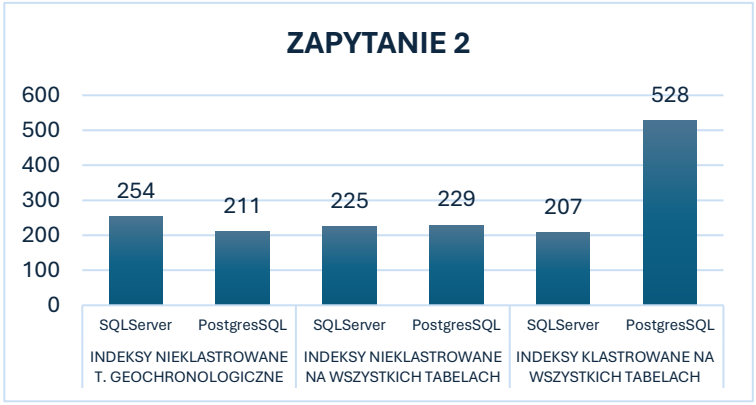
Tabela przedstawia zarówno minimalny czas wykonania zapytania jak i średnią obliczoną z 5 pomiarów. Czas zapytania jest podany w milisekundach.

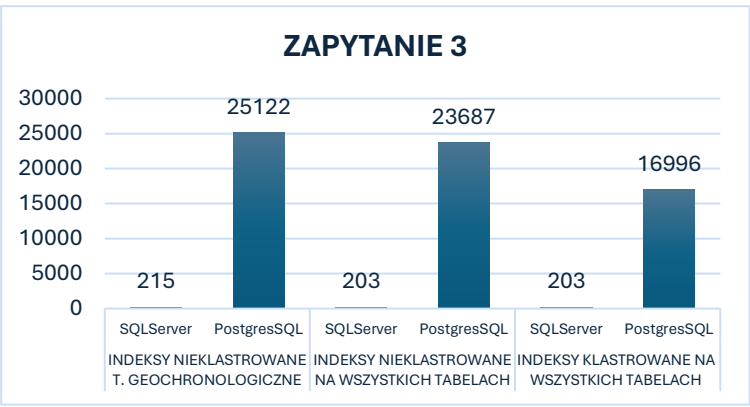
Do obliczenia czasu zapytania użyto komend:
- SQLServer: set statistics time on, set statistics time off
- PostgresSQL: explain analyze

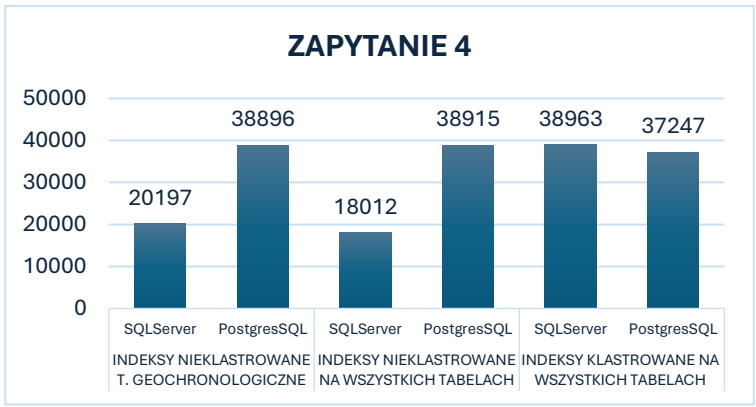
W pierwszym przypadku wykorzystano „cpu time”, a w drugim „execution time”.

Wykresy przedstawiają współzależności między rodzajem bazy danych, typem indeksu a średnim czasem wykonania zapytania.









5. Wnioski

Otrzymane wyniki pozwalają na wyciągnięcie podanych wniosków, związanych z pytaniami postawionymi we wprowadzeniu:

- Zagnieżdżenia skorelowane są wolniejsze niż złączenia, z czego zagnieżdzenia skorelowane schematu znormalizowanego są najmniej wydajne (zapytanie 4).
- Postać zdenormalizowana jest szybsza w obu przypadkach, zarówno dla złączeń jak i zagnieżdżeń,
- W PostgresSQL złączenie dla postaci zdenormalizowanej wykonuje się o wiele szybciej. Średnia zapytania 1 dla postgresa wynosi 99.3, a dla sqlserver 227. Odwrotna sytuacja występuje dla zapytania 3 (schemat zdenormalizowany, zagnieżdżenie skorelowane), gdzie zapytanie wykonuje się średnio o około 20 s dłużej w PostgresSQL niż SQLServer.
- SQLServer jest lepiej zoptymalizowany do wykonywania zagnieżdżeń skorelowanych. Zarówno dla postaci zdenormalizowanej jak i znormalizowanej zapytania wykonywały się szybciej.
- Poziom normalizacji danych w SQL Server nie ma generalnie wpływu na wydajność złączeń. Natomiast dla zagnieżdżeń normalizacja danych do postaci 3NF, powoduje 100 krotne wydłużenie czasu wykonywania zapytania.
- Dodanie indeksu nieklastrowanego do tabeli Milion w systemie SQLServer poprawiło wydajność każdego zapytania. Za to użycie indeksów klastrowanych nie zmienia sytuacji w porównaniu z testem drugim. Dla zagnieżdżeń schematów znormalizowanych nawet pogarsza wydajność.
- Dla systemu PostgresSQL indeksy klastrowane poprawiły jedynie szybkość wykonywania zagnieżdżeń skorelowanych.

Podsumowaniem rozważań są wnioski, że zagnieżdżenia skorelowane są wolniejsze niż złączenia, postać zdenormalizowana jest szybsza niż postać znormalizowana, typ indeksowań nie ma znacznego wpływu na wydajność, ale jego występowanie już tak.