

Sprawozdanie

Testy wydajności

Cel testów:

Porównanie wydajności złączeń i zapytań zagnieżdżonych, wykonanych na tabelach o dużej liczbie danych, w wersji zdenormalizowanej i znormalizowanej.

Sprzęt oraz programy:

Testy zostały wykonane komputerze o parametrach:

CPU: Intel Core i5-8265U, 1.6 GHz, 1800 MHz

RAM: Pamięć DDR4 8 GB (2400 MHz)

SSD: 237 GB

S.O.: Windows 10

Wybrane systemy zarządzania bazami danych to:

SQLServer, wersja 18.11.1

PostgreSQL, wersja 14.3-1

MySQL, wersja Community 8.0.29.0

Testy wykonano kilkakrotnie dla każdego programu. W trakcie testów na komputerze zainstalowane były dwa programy jednocześnie 1) SQLServer i PostgreSQL oraz 2) SQLServer i MySQL.

Przygotowanie:

Do testów wykorzystano tabelę stratygraficzną, skonstruowaną na dwa sposoby:

- tworząc każdą tabelę oddzielnie (schemat znormalizowany)
- tworząc jedną tabelę, zawierającą wszystkie dane (schemat zdenormalizowany):

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

(W SQLServer wykorzystano LEFT JOIN)

Utworzono tabelę *Dziesiec*, na podstawie której utworzono tabelę *Milion* (wypełnioną liczbami od 0 do 999 999), którą łączono z danymi z tabeli stratygraficznej.

cyfra	bit
0	0
1	1
2	0
3	1
4	0
5	1
6	0
7	1
8	0
9	1

Tabela *Dziesiec*:

liczba	cyfra	bit
0	0	0
1	1	1
2	2	0
3	3	1
4	4	0
5	5	1
6	6	0
7	7	1
8	8	0
9	9	1
10	0	0
11	1	1
12	2	0

Tabela *Milion*:

... 999 999

Testy:

Wykonano cztery zapytania sprawdzające wydajność złączeń i zagnieżdżeń z tabelą stratygraficzną w wersji znormalizowanej i zdenormalizowanej. Procedurę przeprowadzono w dwóch etapach:

- Etap I: zapytania bez indeksów na danych
- Etap II: zapytania z indeksami na wszystkich kolumnach biorących udział w złączeniu.

Zapytania:

- 1 ZL - złączenie tablicy *Milion* z tabelą stratygraficzną (zdenormalizowaną), gdzie do warunku złączenia dodano operację modulo, dopasowującą zakresy wartości złączanych kolumn

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

- 2 ZL - złączenie tablicy *Milion* z tabelą stratygraficzną (znormalizowaną), reprezentowaną przez złączenia pięciu tabel

```
--2 ZL
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 ZG - złączenie tablicy *Milion* z tabelą stratygraficzną (zdenormalizowaną), gdzie złączenie jest wykonywane przez zagnieżdżenie skorelowane

```
--3 ZG
SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,68)=
(SELECT id_pietro FROM GeoTabela WHERE mod(Milion.liczba,68)=(id_pietro));
```

- 4 ZG - złączenie tablicy *Milion* z tabelą stratygraficzną (znormalizowaną), gdzie złączenie jest wykonywane przez zagnieżdżenie skorelowane, a zapytanie wewnętrzne jest złączeniem tabel poszczególnych jednostek geochronologicznych

```
--4 ZG
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);
```

Wyniki testów:

Każdy test został przeprowadzony pięciokrotnie, pominięto wyniki skrajne. Wyniki testów zamieszczono w *Tabela 1*.

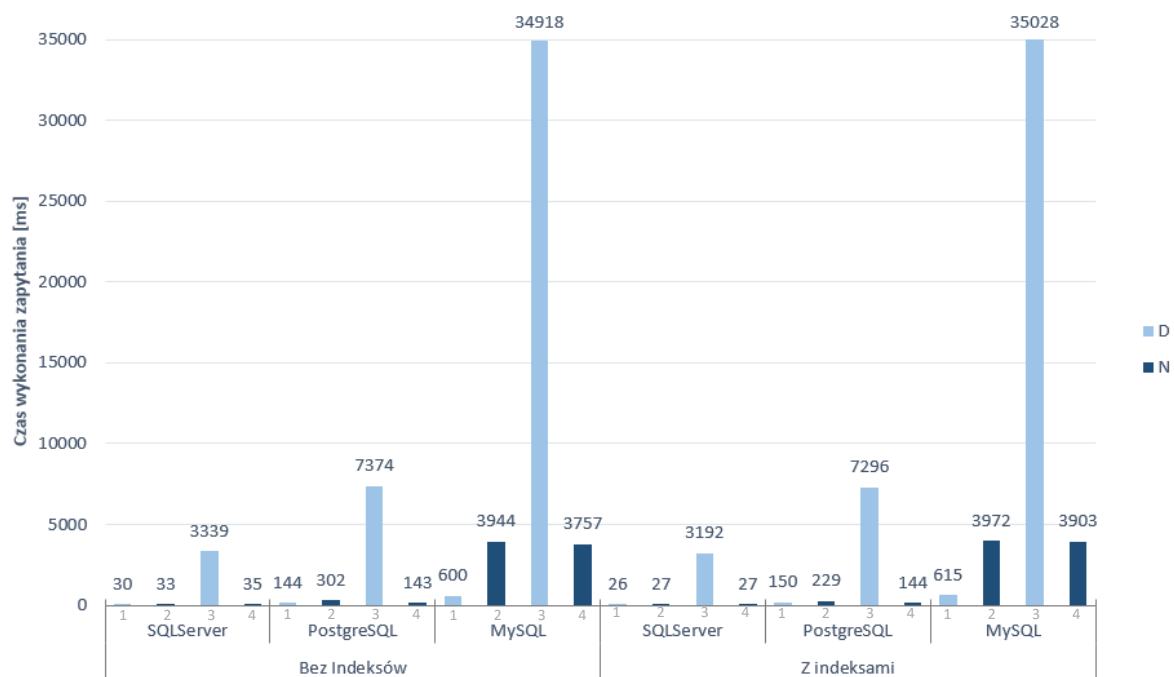
Czasy wykonania zapytań 1 ZL, 2 ZL, 3 ZG i 4 ZG [ms]

	1 ZL (D)		2 ZL (N)		3 ZG (D)		4 ZG (N)	
BEZ INDEKSÓW	MIN	ŚR	MIN	ŚR	MIN	ŚR	MIN	ŚR
SQLServer	26	30	32	33	3213	3339	31	35
PostgreSQL	134	144	293	302	7257	7374	138	143
MySQL	593	600	3891	3944	34390	34918	3750	3757
Z INDEKSAMI								
SQLServer	24	26	26	27	3089	3192	26	27
PostgreSQL	139	150	220	229	7211	7296	137	144
MySQL	593	615	3937	3972	34157	35028	3844	3903

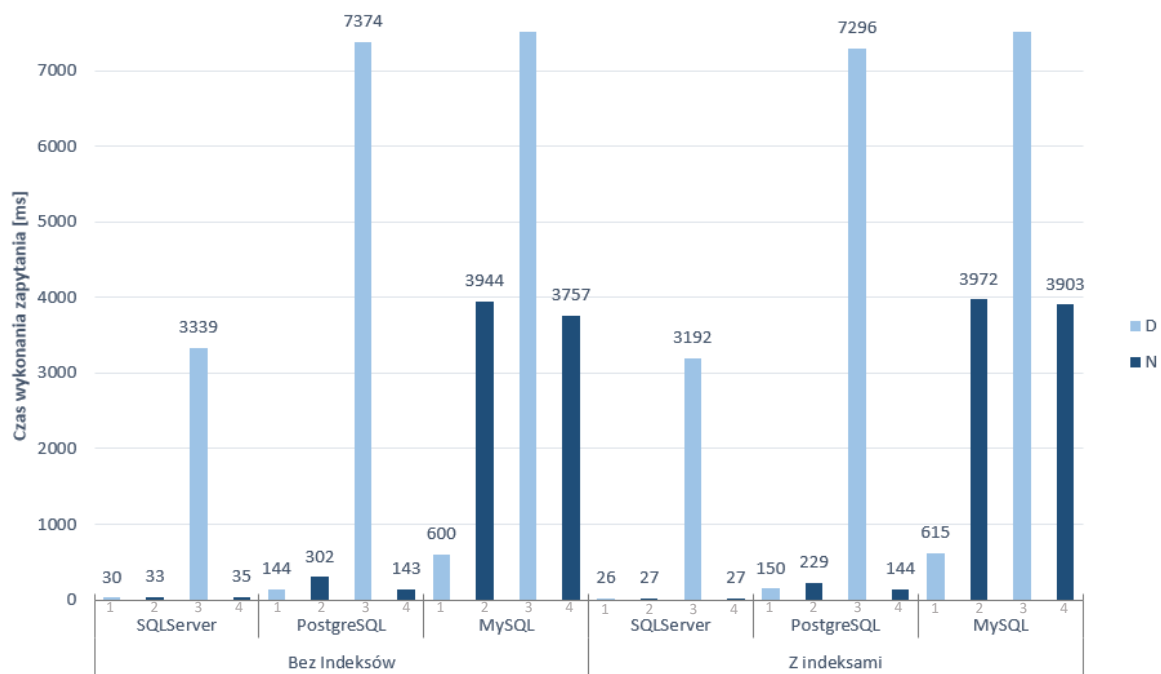
Tabela 1

Dla ułatwienia analizy wyników utworzono wykresy (*Wykres 1* i *Wykres 2*). *Wykres 1* został przedstawiony z pełną skalą liniową. Natomiast *Wykres 2* z częściową skalą liniową, aby ułatwić

porównanie małych wartości. Na wykresach zaznaczono wersję znormalizowaną (N) oraz zdenormalizowaną (D).



Wykres 1



Wykres 2

Wnioski:

- Postać zdenormalizowana jest w większości przypadków wydajniejsza. Przypadek, kiedy postać znormalizowana jest wydajniejsza, to zapytanie 3 ZG, czyli zagnieżdżenie skorelowane. Dłuższy czas wykonywania polecenia występuje dla każdego programu.
- Złączenia są szybsze w wykonaniu niż zagnieżdżenia skorelowane. Wyjątkiem jest złączenie 2 ZL w programie PostgreSQL oraz MySQL, które jest wolniejsze od zagnieżdżenia 4 ZG. Jednak różnica jest tutaj niewielka.
- Zastosowanie indeksów w programie SQLServer we wszystkich przypadkach przyspiesza wykonanie zarówno złączeń, jak i zagnieżdżeń skorelowanych. W MySQL we wszystkich zapytaniach czas wykonywania nieco się zwiększył. Natomiast w systemie PostgreSQL dla zapytania 1 ZL czas lekko się zwiększył, dla zapytań 2 ZL i 3 ZG zmniejszył, a dla zapytania 4 ZG pozostał taki sam. Jednak trzeba brać pod uwagę, że czas jest mierzony w milisekundach, więc w niektórych przypadkach różnice będą znikome.
- Złączenia w SQLServer wykonywane są tak samo szybko dla postaci zdenormalizowanej i znormalizowanej. Inaczej wygląda sytuacja przy zagnieżdżeniu skorelowanym, gdzie czas wykonania jest różny. Zagnieżdżenie dla postaci zdenormalizowanej wykonywane jest o wiele dłużej niż dla postaci znormalizowanej.
- Jak widać najwięcej czasu na wykonanie poleceń potrzebuje program MySQL, a najmniej SQLServer.

Podsumowując, normalizacja w większości przypadków prowadzi do spadku wydajności. Jednak korzyści jakie niesie ze sobą (np. likwidacja problemu powtarzania danych i zmniejszenie ryzyka ich niespójności, optymalizacja objętości bazy danych, łatwość odczytu danych), rekompensują spadek wydajności.