

1. Wstęp

Niniejsze sprawozdanie skupia się na analizie wydajności wykonywania złączeń oraz zapytań zagnieżdżonych na tabelach o dużej liczbie danych. Testy wykonano na jednej maszynie z zamieszczoną niżej konfiguracją sprzętową. W celu zwiększenia wiarygodności wyników badanie przeprowadzono na dwóch systemach zarządzania bazą danych.

2. Konfiguracja sprzętowa

- CPU: Intel(R) Core(™) i5-8350U CPU @ 1.70GHz 1.90GHz
- RAM: DDR3 16GB
- SSD: TOSHIBA 256GB
- S. O.: Windows 10 Pro x64

Jako systemy zarządzania bazami danych wybrano oprogramowanie wolno dostępne:

- PostgreSQL, wersja 13.2
- SQL server, wersja 15.0.2000.5

3. Wykonanie

Stworzyłem bazę danych o nazwie 'GeoBaza' na podstawie artykułu: *WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŻDŻEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH.*

Następnie dziesięciokrotnie wykonałem każde z czterech zapytań, a czasy ich wykonania zapisałem w arkuszu kalkulacyjnych:

➤ 1ZL

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

➤ 2ZL

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

➤ 3ZL

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

➤ 4ZL

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

Kolejnym etapem było nałożenie indeksów na wszystkie kolumny biorące udział w złączeniu oraz ponowne wykonanie wyżej wymienionych zapytań:

```
CREATE INDEX idx_eon ON geo.GeoEon (id_eon);
CREATE INDEX idx_era ON geo.GeoEra (id_era);
CREATE INDEX idx_okres ON geo.GeoOkres (id_okres);
CREATE INDEX idx_epoka ON geo.GeoEpoka (id_epoka);
CREATE INDEX idx_pietro ON geo.GeoPietro (id_pietro);
CREATE INDEX idx_milion ON geo.milion (liczba);
CREATE INDEX idx_dziesiec ON geo.dziesiec (cyfra);
```

4. Wyniki

Czasy wykonania w PostgreSQL bez nałożonych indeksów:

1ZL [ms]	2ZL [ms]	3ZL [ms]	4ZL [ms]
205	365	11441	226
222	370	12178	215
200	389	11458	241
209	371	11448	221
217	385	11451	186
223	421	11618	231
253	380	11444	210
233	412	11467	232
211	351	11333	261
223	380	12190	242

Średnie:

1ZL [ms]	2ZL [ms]	3ZL [ms]	4ZL [ms]
219,6	382,4	11602,8	226,5

Czasy wykonania w PostgreSQL z nałożonymi indeksami:

1ZL [ms]	2ZL [ms]	3ZL [ms]	4ZL [ms]
220	282	11389	226
222	348	11468	208
228	349	11462	187
234	286	11445	214
188	326	11534	230
222	351	11500	211
194	271	11482	204
227	329	11489	193
201	264	11468	206
224	305	11423	224

Średnie:

1ZL [ms]	2ZL [ms]	3ZL [ms]	4ZL [ms]
216	311,1	11466	210,3

Czasy wykonania w SQL Server bez nałożonych indeksów:

1ZL [ms]	2ZL [ms]	3ZL [ms]	4ZL [ms]
305	294	285	279
293	287	352	284
274	288	308	281
269	296	277	280
283	311	292	304
283	289	272	268
282	289	273	341
276	280	280	277
307	279	303	274
316	291	270	280

Średnie:

1ZL [ms]	2ZL [ms]	3ZL [ms]	4ZL [ms]
288,8	290,4	291,2	286,8

Czasy wykonania w SQL Server z nałożonymi indeksami:

1ZL [ms]	2ZL [ms]	3ZL [ms]	4ZL [ms]
283	257	283	274
298	262	318	306
266	250	283	301
283	283	283	268
299	265	283	266
284	296	298	273
298	265	282	266
266	286	298	290
267	281	271	283
314	265	266	263

Średnie:

1ZL [ms]	2ZL [ms]	3ZL [ms]	4ZL [ms]
285,8	271	286,5	279

5. Wnioski

Bazując na uzyskanych wynikach możemy zauważyć, iż po nałożeniu indeksów wykonanie zapytań oraz złączeń zagnieżdżonych skróciło się. Jednakże, średnie czasy zmniejszyły się jedynie o kilka milisekund co nie może jednoznacznie potwierdzić efektywności indeksacji. Może to być spowodowane działaniem różnych procesów wykonywanych przez system operacyjny w trakcie wywoływania zapytań oraz niestabilną wydajnością laptopa, na którym cały test był wykonywany.