1. Wstęp

Niniejsza analiza miała na celu zbadanie i porównanie wydajności zapytań opartych na złączeniach i zagnieżdżeniach w przypadku tabeli geologicznej.

Badanie zostało przeprowadzone za pomocą systemu PostreSQL.

2. Konfiguracja sprzętowa

- CPU: AMD Ryzen 5 5500
- GPU: GAINWARD RTX 3060
- RAM: 16 GB
- OS: WINDOWS 10 HOME

3. Narzędzie pracy

- PostgreSQL-15.3-1-WINDOWS-X64

4. Zapytania testowe

Zapytanie 1 miało na celu połączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci zdenormalizowanej.

```
SELECT
    COUNT(*)
FROM
    liczby.milion m
INNER JOIN
    tabela_stratygraficzna.TabelaStr t
ON
    m.liczba%95=CAST(RIGHT(t.ID_pietra, LENGTH(t.ID_pietra)-3) AS INT);
```

Zapytanie 2 miało na celu połączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci znormalizowanej.

```
SELECT
        COUNT(*)
FROM
        liczby.milion m
INNER JOIN
        tabela_stratygraficzna.GeoPietro geo_p
ON
        m.liczba%95=CAST(RIGHT(geo_p.id_pietro, LENGTH(geo_p.id_pietro)-3) AS INT)
INNER JOIN
        tabela_stratygraficzna.GeoEpoka geo_ep
ON
        geo_p.id_epoka=geo_ep.id_epoka
INNER JOIN
        tabela_stratygraficzna.GeoOkres geo_o
ON
        geo_ep.id_okres=geo_o.id_okres
INNER JOIN
        tabela_stratygraficzna.GeoEra geo_er
ON
        geo_o.id_era=geo_er.id_era
INNER JOIN
        tabela_stratygraficzna.GeoEon geo_eo
ON
        geo_er.id_eon=geo_eo.id_eon;
```

Zapytanie 3 ma na celu połączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w zdenormalizowanej postaci, przy czym połączenie jest realizowane poprzez zagnieżdżenie skorelowane.

```
SELECT
    COUNT(*)
FROM
    liczby.milion m
WHERE
    m.liczba%95=
    (SELECT
        CAST(RIGHT(t.ID_pietra, LENGTH(t.ID_pietra)-3) AS INT)
    FROM
        tabela_stratygraficzna.TabelaStr t
    WHERE
        m.liczba%95=CAST(RIGHT(t.ID_pietra, LENGTH(t.ID_pietra)-3) AS INT));
```

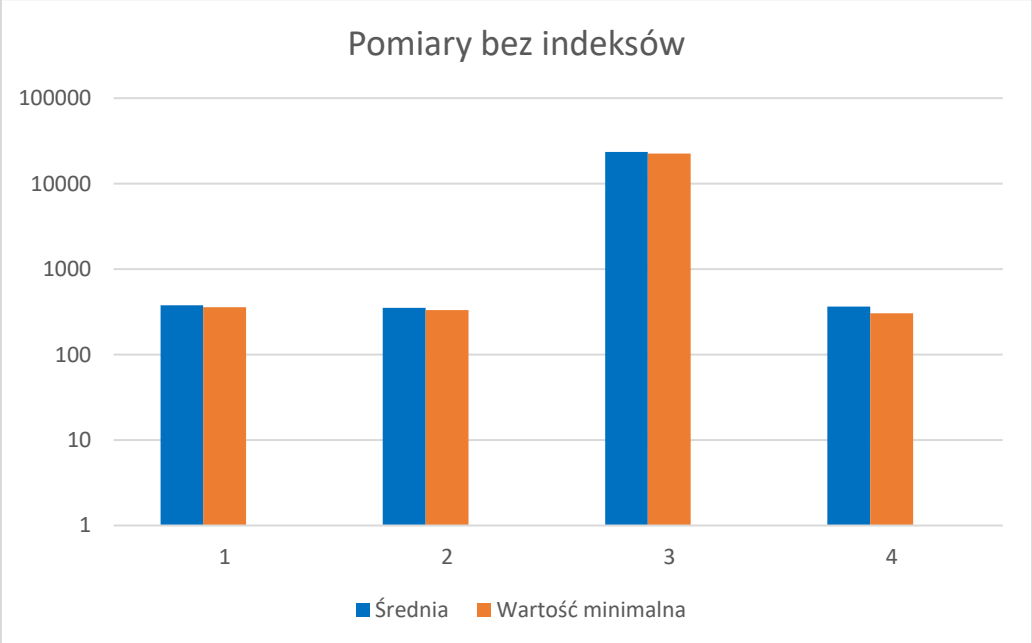
Zapytanie 4 ma na celu połączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w znormalizowanej postaci. Połączenie to jest realizowane poprzez zagnieżdżenie skorelowane, przy czym zapytanie wewnętrzne obejmuje łączenie poszczególnych jednostek geochronologicznych z tabel.

```
SELECT
    COUNT(*)
FROM
    liczby.milion m
WHERE
    m.liczba%95 IN
    (SELECT
        CAST(RIGHT(geo_p.id_pietro, LENGTH(geo_p.id_pietro)-3) AS INT)
    FROM
        tabela_stratygraficzna.GeoPietro geo_p
    INNER JOIN
        tabela_stratygraficzna.GeoEpoka geo_ep
```

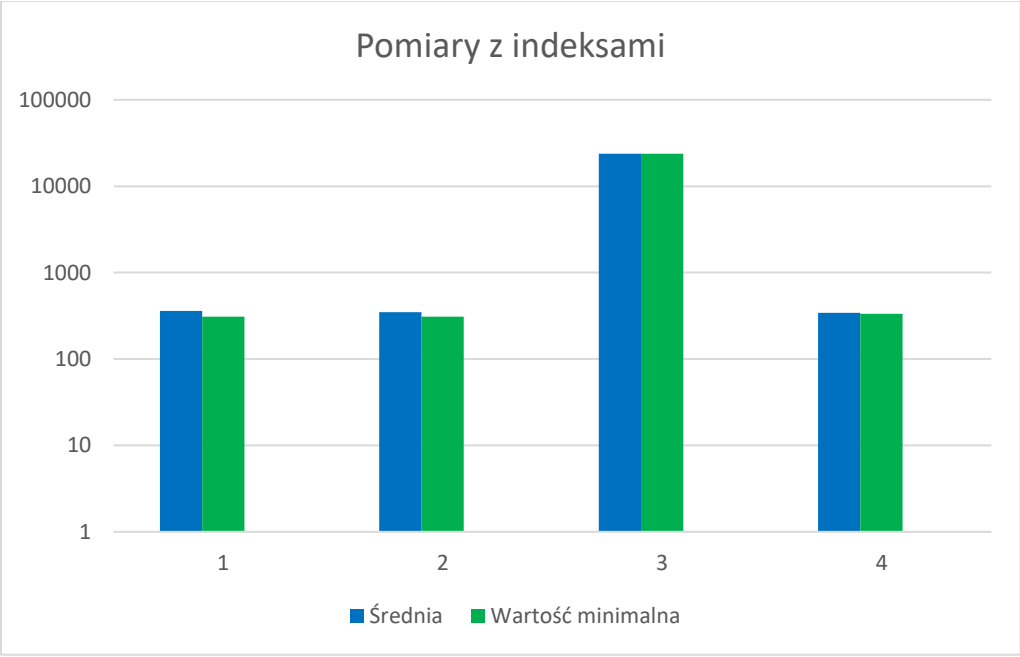
```
ON
        geo_p.id_epoka=geo_ep.id_epoka
INNER JOIN
        tabela_stratygraficzna.GeoOkres geo_o
ON
        geo_ep.id_okres=geo_o.id_okres
INNER JOIN
        tabela_stratygraficzna.GeoEra geo_er
ON
        geo_o.id_era=geo_er.id_era
INNER JOIN
        tabela_stratygraficzna.GeoEon geo_eo
ON
        geo_er.id_eon=geo_eo.id_eon);
```

5. Wyniki testów

	Pomiary bez indeksów			
	1	2	3	4
	371	351	23 780	385
	386	343	23690	392
	392	365	23567	366
	357	355	23954	305
	375	338	23321	389
	381	360	23564	382
	396	366	23332	317
	358	332	23754	334
	388	349	23645	397
	365	361	22456	371
Średnia	376,9	352	23 506	363,8



	Pomiary z indeksami			
	1	2	3	4
	364	355	23831	356
	394	396	23849	341
	399	308	23862	339
	347	344	23892	334
	318	363	23803	337
	308	387	23878	338
	384	312	23878	341
	371	330	23838	340
	315	319	23897	354
	397	372	23811	343
Średnia	359,7	348,6	23854	342,3



6. Wnioski

Na podstawie powyższych wyników można stwierdzić, że:

- Indeksacja poprawiła wydajność wykonywania zapytań dla skomplikowanych operacji, takich jak zagnieżdżanie, natomiast dla małych tabel i prostych zapytań wydłużyła czas wykonania.

- W większości przypadków forma zdenormalizowana jest lepsza.

Ostatecznym wnioskiem jest stwierdzenie, że normalizacja w większości przypadków prowadzi do spadku wydajności, ale umożliwia łatwe przechowywanie danych w zrozumiały sposób, zmniejsza szanse na wystąpienie błędów oraz porządkuje dane.