

# BAZY DANYCH ĆW9

Nadia Maksymiuk

gr 2

nr indeksu: 402683

## 1. Cel projektu

Celem projektu było sporządzenie tabeli stratygraficznej oraz sprawdzenie wydajności złączeń i zagnieżdżeń dla schematów skorelowanych i nieskorelowanych.

## 2. Parametry sprzętu i programu

Przy realizacji ćwiczenia pracowałam z programem MySQL Workbench 8.0 CE

Pracowałam na komputerze z danymi podzespołami:

- Procesor Intel Core i7-7700HQ 2.8GHz 4-rdzenie
- Pamięć RAM 16 GB
- System operacyjny Microsoft Windows 10 Home
- Lenovo Legion Y520

## 3. Testy

Wykonałam 12 zapytań sprawdzających wydajność złączeń i zagnieżdżeń dla tabeli stratygraficznej w wersjach skorelowanej i nieskorelowanej. Wykonałam te zapytania w dwóch etapach. Pierwszy z nich był dla tabel bez nałożonych indeksów natomiast drugi był dla tabel z nałożonymi indeksami na kolumnach.

Zasadniczym celem testów była ocena wpływu normalizacji na zapytania złożone –złączenia i zagnieżdżenia (skorelowane).

W tym celu zaproponowano cztery zapytania:

- **1 ZL**, którego celem było złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci ZDENORMALIZOWANEJ

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoTabela ON  
(mod(Milion.liczba,98) = (GeoTabela.id_pietro));
```

- **2 ZL**, którego celem którego celem było złączenie syntetycznej tablicy miliona wyników z tabelą geochronologiczną w postaci ZNORMALIZOWANEJ reprezentowanej przez złączenia 5 tabel

```
SELECT COUNT(*) FROM Milion INNER JOIN GeoPietro ON
(mod(Milion.liczba,98)=GeoPietro.id_pietro) NATURAL JOIN GeoEpoka
NATURAL JOIN GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon;
```

- **3 ZG**, którego celem jest 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,98)= (SELECT
id_pietro FROM GeoTabela WHERE mod(Milion.liczba,98)=(id_pietro));
```

- **4 ZG**, którego celem 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 tabel poszczególnych jednostek geochronologicznych:

```
SELECT COUNT(*) FROM Milion WHERE mod(Milion.liczba,98) IN (SELECT
GeoPietro.id_pietro FROM GeoPietro NATURAL JOIN GeoEpoka NATURAL JOIN
GeoOkres NATURAL JOIN GeoEra NATURAL JOIN GeoEon;
```

#### 4. Wyniki

Każdy z testów przeprowadziłam 12 razy. Z testów wyróżniłam średnią oraz minimalną wartość. Dane zamieściłam w tabeli poniżej:

	1 ZL		2 ZL		3 ZG		4 ZG	
	MIN	ŚR	MIN	ŚR	MIN	ŚR	MIN	ŚR
Bez indeksów	750	792	797	840	60406	63301	2547	3043
Z indeksami	2125	2454	2313	2639	2938	3336	2344	2458

## 5. Wnioski

Z podanych wyników można wyciągnąć dane wnioski:

- Użycie indeksów w systemie MySQL we wszystkich testach przyspiesza wykonanie zapytań dla zagnieżdżeń i złączeń.
- Zagnieżdżenia skorelowane są dużo wolniejsze w wykonaniu niż złączenia.
- Zagnieżdżenie skorelowane zdenormalizowanej wersji tabeli sprawiło programowi MySQL najwięcej problemów