

WYDAJNOŚĆ ZŁĄCZEŃ I ZAGNIEŹDZEŃ DLA SCHEMATÓW ZNORMALIZOWANYCH I ZDENORMALIZOWANYCH

1. TABELA GEOCHRONOLOGICZNA

Jest to schemat obrazujący przebieg historii Ziemi. Stworzono ją na podstawie następstwa procesów geologicznych, a tym samym układu warstw skalnych. Poniżej przedstawiona jest tabela ustalona przez Międzynarodową Komisję Stratygrafii (ICS).

W tabeli przedstawiono taksonomię dla pięciu jednostek geochronologicznych - eonu, ery, okresu, epoki i piętra.

eon	era	okres	epoka	piętro
Fanerozoik			Holocen	11,5 tys
Kenozoik		Neogen	Plejstocen	Górny 126 tys.
				Środkowy 781 tys.
				Dolny 1,806 mln
			Pliocen	gelas 2,588 mln
				piacent 3,6 mln
				zankl 5,332 mln
			Miocen	messyn 7,246 mln
				torton 11,608 mln
				serrawal 13,65 mln
				lang 15,97 mln
		Paleogen	Oligocen	burdygal 20,43 mln
				akwitan 23,03 mln
				szat 28,4 (± 0,1) mln
			Eocen	rupel 33,9 (± 0,1) mln
				priabon 37,2 (± 0,1) mln
				barton 40,4 (± 0,2) mln
			Paleocen	lutet 48,6 (± 0,2) mln
				iprez 55,8 (± 0,2) mln
				tanet 58,7 (± 0,2) mln
				seland 61,7 (± 0,2) mln
Mezozoik		Kreda	Górna Kreda	dan 65,5 (± 0,3) mln
				mastrycht 70,6 (± 0,6) mln
				kampan 83,5 (± 0,7) mln
				santon 85,8 (± 0,7) mln
				koniak 89,3 (± 1,0) mln
				turon 93,5 (± 0,8) mln
				cenoman 99,6 (± 0,9) mln

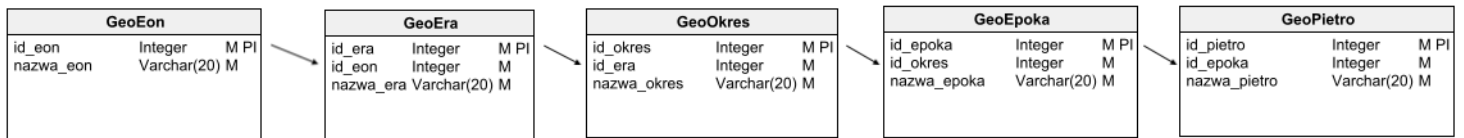
Jura	Dolna Kreda	alb 112 (± 1,0) mln
		apt 125 (± 1,0) mln
		barrem 130 (± 1,5) mln
		hoteryw 136,4 (± 2,0) mln
		walanżyn 140,2 (± 3,0) mln
		berias 145,5 (± 4,0) mln
	Górna Jura	kimeryd 155,7 (± 4,0) mln
		oksford 161,2 (± 4,0) mln
		kelowej 164,7 (± 4,0) mln
		baton 167,7 (± 3,5) mln
	Środkowa Jura	bajos 171,6 (± 2,0) mln
		aalen 175,6 (± 2,0) mln
		toark 183 (± 1,5) mln
		pliensbach 189,6 (± 1,5) mln
Trias	Dolna Jura	synemur 196,5 (± 1,0) mln
		hetang 199,6 (± 0,6) mln
	Górny Trias	retyk 203,6 (± 1,5) mln
		noryk 216,5 (± 2,0) mln
		karnik 228 (± 2,0) mln
	Środkowy Trias	ladyn 237 (± 2,0) mln
		anizyk 245 (± 1,5) mln
	Dolny Trias	olenek 249,7 (± 0,7) mln
		ind 251,0 (± 0,4) mln

Paleozoik	Perm	Loping	szangsing 253,8 (± 0,7) mln
			wucziaping 260,4 (± 0,7) mln
		Gwadelup	kapitan 265,8 (± 0,7) mln
			word 268 (± 0,7) mln
		Cisural	road 270,6 (± 0,7) mln
			kungur 275,6 (± 0,7) mln
			atryńsk 284,4 (± 0,7) mln
			sakmar 294,6 (± 0,8) mln
		Pensylwan	assel 299 (± 0,8) mln
			gźel 303,9 (± 0,9) mln
Karbon	Pensylwan	Górny Pensylwan	kazim 306,5 (± 1,0) mln
			moskow 311,7 (± 1,1) mln
		Dolny Pensylwan	bakszir 318,1 (± 1,3) mln
			serpuchow 326,4 (± 1,6) mln
	Missisip	Górny Missisip	wizen 354,3 (± 2,1) mln
			turnej 359,2 (± 2,5) mln
		Środkowy Missisip	
		Dolny Missisip	
Dewon		Górny Dewon	famen 374,5 (± 2,6) mln
			fran 385,3 (± 2,6) mln
		Środkowy Dewon	żywet 391,8 (± 2,7) mln
			eifel 397,5 (± 2,7) mln
		Dolny Dewon	ems 407 (± 2,8) mln
			prag 411,2 (± 2,8) mln
			loczkow 416 (± 2,8) mln

Rys.1 Tabela geochronologiczna

2. KONSTRUKCJA WYMIARU GEOCHRONOLOGICZNEGO

W celu badań wydajności złączeń i zagnieżdżeń stworzono początkowo znormalizowany schemat tabeli geochronologicznej, który przedstawia się następująco:



Rys.2 Znormalizowany schemat tabeli geochronologicznej



Dzięki utworzeniu jednej tabeli o nazwie „GeoTabela”, która to zawiera wszystkie dane ze znormalizowanego schematu tabeli, łącząc je za pomocą złączenia naturalnego, stworzono także jej schemat zdenormalizowany przedstawiony poniżej:

GeoTabela		
id_pietro	int	PK
nazwa_pietro	varchar(20)	
id_epoka	int	
nazwa_epoka	varchar(20)	
id_okres	int	
nazwa_okres	varchar(20)	
id_era	int	
nazwa_era	varchar(20)	
id_pietro	int	
nazwa_pietro	varchar(20)	



Rys.3 Zdenormalizowany schemat tabeli geochronologicznej

```

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

Rys.4 Zapytanie tworzące tabelę „GeoTabela”

Dzięki temu zabiegowi, otrzymano szybki dostęp do wszystkich danych tabeli geochronologicznej

3. TESTY WYDAJNOŚCI

Przeprowadzono testy na trzech różnych darmowych rozwiązaniach bazodanowych:

MySQL

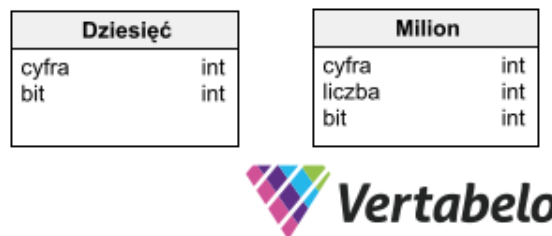
PostgreSQL

Skupiono się na porównaniu wydajności zapytań zagnieżdżonych, oraz złączeń, które to zostały wykonane na tabelach o dużej liczbie danych.

Aby wykonać testy wykonano 4 różne zapytania.

W owych zapytaniach połączono dane z tabeli geochronologicznej, z danymi zawartymi w nowo stworzonej tabeli Milion.

Tabela milion, to tabela powstała za pomocą odpowiedniego autozłączenia tabeli o nazwie Dziesięć, wypełnionej liczbami od 0 do 9.



Rys.5 Schemat tabel Dziesięć i Milion

```
CREATE TABLE Stratygrafia.Milion
(
    Liczba INT,
    Cyfra INT,
    Bit INT
);
INSERT INTO Stratygrafia.Milion
SELECT
a1.cyfra +10* a2.cyfra +100*a3.cyfra + 1000*a4.cyfra+ 10000*a5.cyfra + 10000*a6.cyfra AS liczba,
a6.cyfra AS cyfra,
a6.bit AS bit
FROM
Stratygrafia.Dziesiec a1,
Stratygrafia.Dziesiec a2,
Stratygrafia.Dziesiec a3,
Stratygrafia.Dziesiec a4,
Stratygrafia.Dziesiec a5,
Stratygrafia.Dziesiec a6;
```

Rys.6 Tworzenie tabeli milion za pomocą autozłączenia

3.1 Konfiguracja sprzętowa i programowa

Wszystkie testy omówione w niniejszym sprawozdaniu wykonano na urządzeniu o następujących parametrach:

CPU: Apple M1 - 8-core CPU

RAM: M1 Pro chip with 8GB of RAM

HDD: 512 GB

S.O.: macOS Big Sur 11.1

MySQL: wersja 8.0.29

PostgreSQL: wersja 6.4 (4280.88)

Testy wykonywano wielokrotnie na tym samym komputerze, posiadającym powyższe parametry.

3.2 Kryteria testów

Przeprowadzono 4 testy, powielając je 5 razy.

Sprawdzały one wydajność zagnieżdżeń i złączeń z tabelą geochronologiczną w wersji zdenormalizowanej i znormalizowanej.

Procedura obejmowała dwa etapy:

1. Zapytania nie miały nałożonych indeksów na kolumny danych
2. Na zapytania nałożono indeksy na wszystkie kolumny biorące udział w złączeniu.

Poniżej opisuje się zapytania będące przedmiotem testów:

1. ZAPYTANIE 1 (1 ZL)

Celem zapytania jest złączenie tablicy Milion z zdenormalizowaną tabelą geochronologiczną. Do warunku złączenia dodano operację modulo, która dopasowywała zakresy wartości złączanych kolumn.

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

2. ZAPYTANIE 2 (2 ZL)

Celem zapytania jest złączenie tablicy Milion z znormalizowaną tabelą geochronologiczną.

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

3. ZAPYTANIE 3 (3 ZG)

Celem zapytania jest złączenie tablicy Milion z zdenormalizowaną tabelą geochronologiczną. Złączenie jest tutaj wykonywane przez skorelowane zagnieżdżenie.

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

4. ZAPYTANIE 4 (4 ZG)

Celem zapytania jest złączenie tablicy Milion z znormalizowaną tabelą geochronologiczną. Zapytanie wewnętrzne jest złączeniem tabel poszczególnych jednostek geochronologicznych. Złączenie jest tutaj wykonywane przez skorelowane zagnieżdżenie.

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

5. Wyniki testów

Każdy z testów przeprowadzono wielokrotnie, a dokładnie zrobiono 5 prób dla każdego zapytania. W oprogramowaniu MySQL kolejne próby były do siebie niezwykle zbliżone. Wyniki testów zamieszczone zostały w poniższej tabeli

					PostgreSQL							
	BEZ INDEKSÓW							Z INDEKSAMI				
	PRÓBA 1	PRÓBA 2	PRÓBA 3	PRÓBA 4	PRÓBA 5			PRÓBA 1	PRÓBA 2	PRÓBA 3	PRÓBA 4	PRÓBA 5
1 ZL	105	112	98	97	95		1 ZL	72	76	71	74	76
2 ZL	178	159	156	174	168		2 ZL	115	124	117	111	114
3 GZ	3613	3563	3581	3642	3615		3 GZ	3550	3540	3548	3545	3543
4 GZ	116	102	155	81	74		4 GZ	79	75	75	76	72
						MySQL						
	BEZ INDEKSÓW							Z INDEKSAMI				
	PRÓBA 1	PRÓBA 2	PRÓBA 3	PRÓBA 4	PRÓBA 5			PRÓBA 1	PRÓBA 2	PRÓBA 3	PRÓBA 4	PRÓBA 5
1 ZL	232	235	241	232	232		1 ZL	230	230	231	230	231
2 ZL	259	258	260	260	257		2 ZL	1314	1316	1323	1311	1308
3 GZ	14346	14360	14390	14402	14381		3 GZ	14284	14330	14355	14381	14384
4 GZ	258	252	258	255	255		4 GZ	1325	1316	1316	1316	1318

Tabela 1. Wyniki każdej z prób

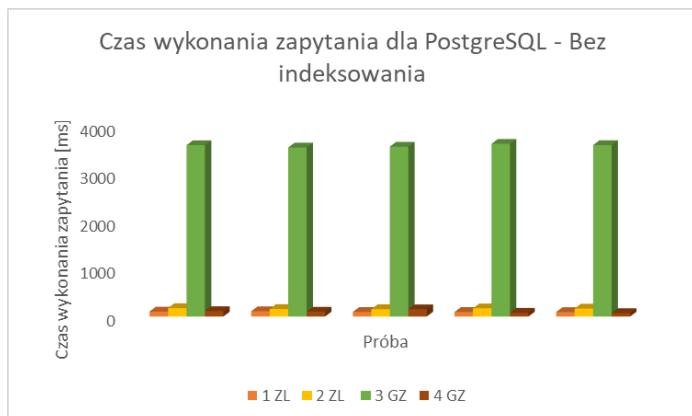
Wyniki przedstawione w tabeli 1 zebrano i przedstawiono w tabeli 2, wyciągając minimalny i średni czas wykonania zapytania.

Czas wykonania zapytań 1 ZL, 2 ZL, 3 ZG, 4ZG

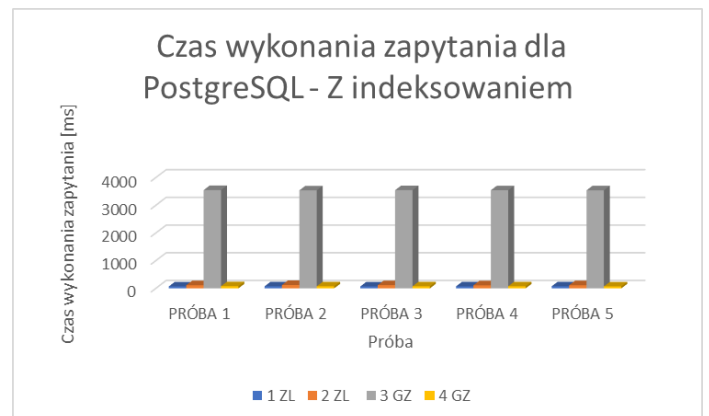
	1 ZL		2 ZL		3 ZG		4 ZG	
	MIN	ŚR	MIN	ŚR	MIN	ŚR	MIN	ŚR
BEZ INDEKSÓW								
PostgreSQL	95	101,4	156	167	3563	3602,8	74	105,6
MYSQL	232	234,4	257	258,8	14346	14375,8	252	255,6
Z INDEKSAMI								
PostgreSQL	71	73,8	111	116,2	3540	3545,2	72	75,4
MYSQL	230	230,4	1308	1314,4	14284	14346,8	1316	1318,2

Tabela 2. Minimalny i średni czas wykonania zapytania dla poszczególnego oprogramowania z uwzględnieniem indeksowania

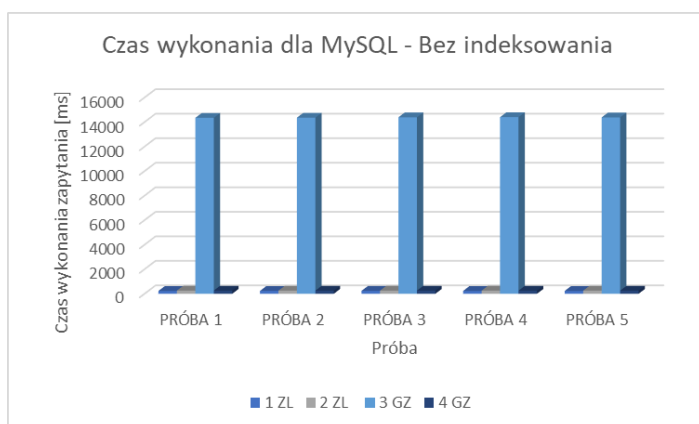
Analizę wyników ułatwiają wizualizacje. W tym przypadku zastosowano cztery wykresy słupkowe.



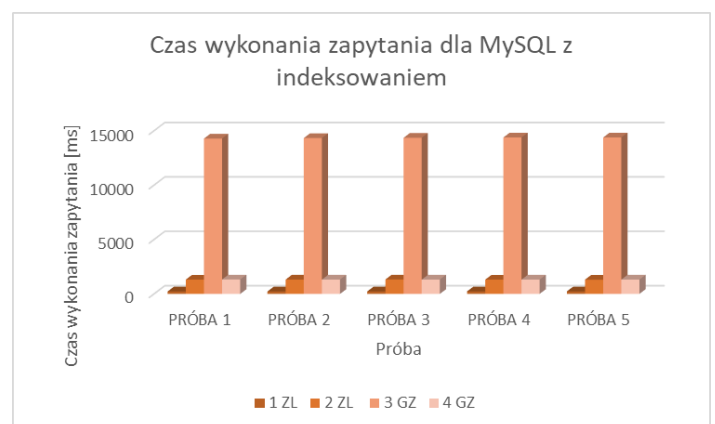
Wykres 1. Wyniki PostgreSQL Bez indeksowania



Wykres 2. Wyniki PostgreSQL z indeksowaniem



Wykres 4. Wyniki MySQL bez indeksowania

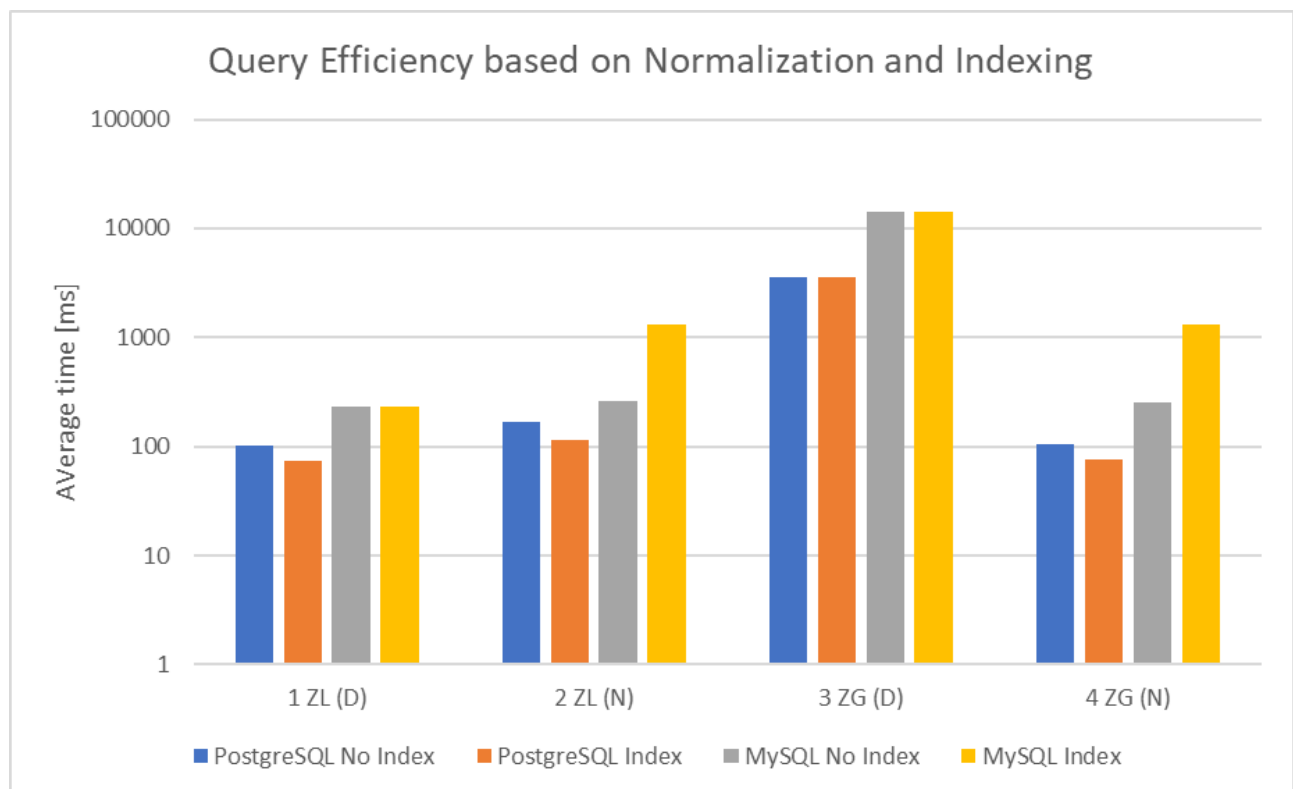


Wykres 5. Wyniki MySQL z indeksowaniem

6. Wnioski

Otrzymane wyniki pozwalają wyciągnąć wnioski:

1. Oprogramowanie MySQL daje wyniki zapytań w bardzo zbliżonym do siebie czasie.
2. Postać zdenormalizowana jest generalnie szybsza niż postać znormalizowana.
3. Złączenia są dużo wydajniejsze czasowo niż zagnieżdżenia, które niezwykle spowalniają wykonywanie zapytań.
4. W oprogramowaniu PostgreSQL, po użyciu indeksowania w każdym przypadku widać przyspieszenie wykonania zapytania.
5. W Oprogramowaniu PostgreSQL, indeksowanie w przypadku zapytania z zagnieżdżeniem 3GZ dało najmniejsze przyspieszenie. Okazało się ono minimalne
6. W oprogramowaniu MySQL, po użyciu indeksowania w przypadku postaci znormalizowanej (zapytania 2 ZL i 4ZG) nastąpiło duże spowolnienie wykonania zapytań.
7. W oprogramowaniu MySQL, po użyciu indeksowania w przypadku postaci zdenormalizowanej (zapytania 1 ZL i 3ZG) nastąpiło minimalne przyspieszenie wykonania zapytań.
8. Najwolniej wykonywanym zapytaniem, zarówno w przypadku użycia oprogramowania MySQL jak i oprogramowania PostgreSQL jest zapytanie 3GZ, czyli zapytanie w postaci znormalizowanej z użyciem złączenia wykonywanego poprzez zagnieżdżenie skorelowane.
9. W przypadku zapytania 3GZ, które jest wykonywane najdłużej ze wszystkich w obu przypadkach, zdecydowanie lepiej radzi sobie z nim oprogramowanie PostgreSQL niż MySQL.
10. W wykonywanych testach wydajniejszym systemem do zarządzania baz danych okazał się PostgreSQL.



Wykres 6. Wyniki MySQL z indeksowaniem

Podsumowując, wprowadzenie indeksów w większości przypadków daje spore przyspieszenie wykonania zapytań. Jedynymi przypadkami w których wykryto spadek wydajności po wprowadzeniu tego zabiegu są zapytania znormalizowane w oprogramowaniu „MySQL”. Generalnie postać zdenormalizowana powoduje wzrost wydajności. Postać znormalizowana prowadzi zaś do jej spadku. Mimo wszystko normalizacja wprowadza spory porządek i umożliwia łatwą konserwację.

Wykonane testy pozwoliły na porównanie dwóch systemów do zarządzania bazami danych. Pokazały one, że w badanych przypadkach, oprogramowaniem reprezentującym lepszą wydajność okazał się system „PostgreSQL”. MySQL może pochwalić się jednak dużą zgodnością czasową kolejnych prób.

BIBLIOGRAFIA

1. www.vertabelo.com
2. <https://notatek.pl/tabela-stratygraficzna-uproszczona-1>
3. https://pl.wikipedia.org/wiki/Tabela_stratygraficzna
4. Łukasz Jajeńnica, Adam Piórkowski - Wydajność złączeń i zagnieżdżeń dla schematów znormalizowanych i zdenormalizowanych. Studia informatica - number 2A