

# RAPORT

## Łukasz Kozłowski

### nr 401715

#### 1. Konfiguracja sprzętowa i programowa

Wszystkie testy zostały przeprowadzone na komputerze o następujących parametrach

CPU: AMD Ryzen 5 2600 Six-Core Processor 3,60GHz

RAM: Pamięć DDR4 16 GB

HDD: Toshiba 7200 rpm, 64 MB OEM

S.O.: Windows 10 Home

Oprogramowania:

Microsoft SQL Server Management Studio wersja 15.0.2000.5

Postgres wersja 4.30

Testy zostały wykonane na komputerze, na którym były zainstalowane obydwie oprogramowania jednocześnie.

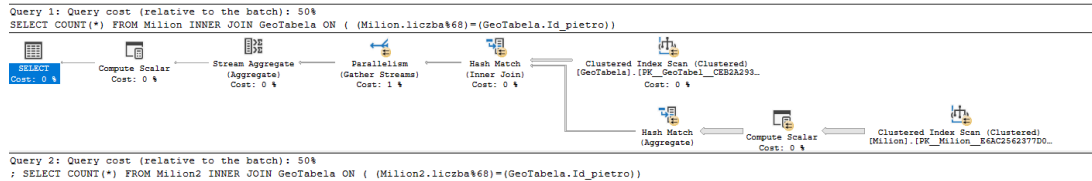
#### 2. Wyniki pomiarów

Sql Server								
	Z indexami [ms]				Bez indexów [ms]			
Lp.	1ZL	2ZL	3ZG	4ZG	1ZL	2ZL	3ZG	4ZG
1	19	23	22	23	22	23	20	30
2	16	23	22	24	18	27	18	24
3	16	25	17	24	18	24	20	29
4	19	28	17	23	21	24	25	31
5	19	23	19	23	18	24	22	24
6	18	24	18	21	21	29	18	36
7	19	23	18	22	20	25	19	29
8	19	21	15	21	22	25	20	26
9	19	24	18	24	21	25	18	27
10	18	21	17	22	18	25	19	29
średnia	18,2	23,5	18,3	22,7	19,9	25,1	19,9	28,5
minimum	16	21	17	21	18	23	18	24

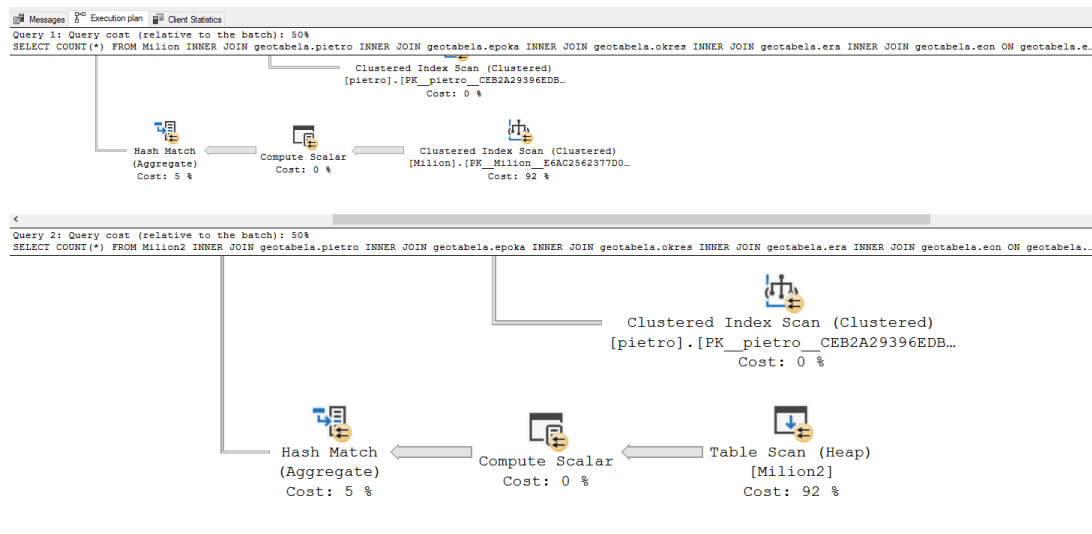
Postgres								
	Z indexami [ms]				Bez indexów [ms]			
Lp.	1ZL	2ZL	3ZG	4ZG	1ZL	2ZL	3ZG	4ZG
1	194	300	10 595	238	177	285	10 608	190
2	247	302	10 652	175	176	278	10 744	192
3	182	306	10 749	171	185	313	10 611	178
4	165	315	10 556	175	187	276	10 640	195
5	202	335	10 727	169	170	379	10 717	185
6	187	314	10 737	206	171	422	10 816	193
7	183	302	10 610	189	174	273	11 419	191
8	243	326	10 669	180	337	281	10 774	237
9	190	324	10 640	180	159	390	11 315	188
10	232	320	10 541	175	302	280	10 687	215
średnia	202,5	314,4	10647,6	185,8	203,8	317,7	10833,1	196,4
minimum	165	314	10541	169	18	23	18	24

### 3. Plan realizacji

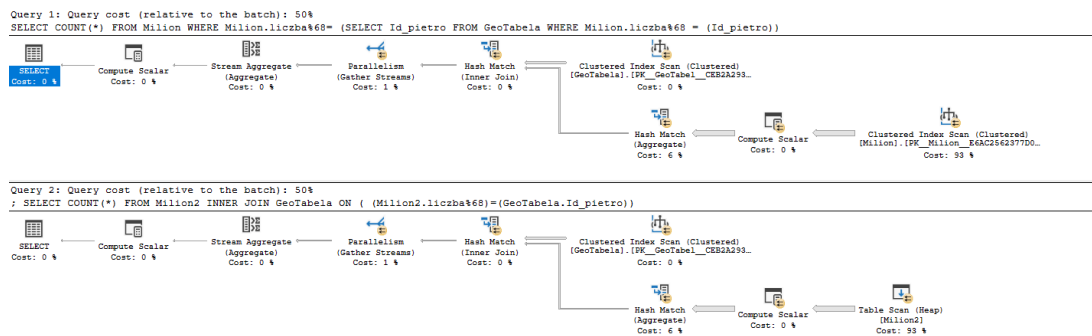
#### Dla Sql Server 1ZL



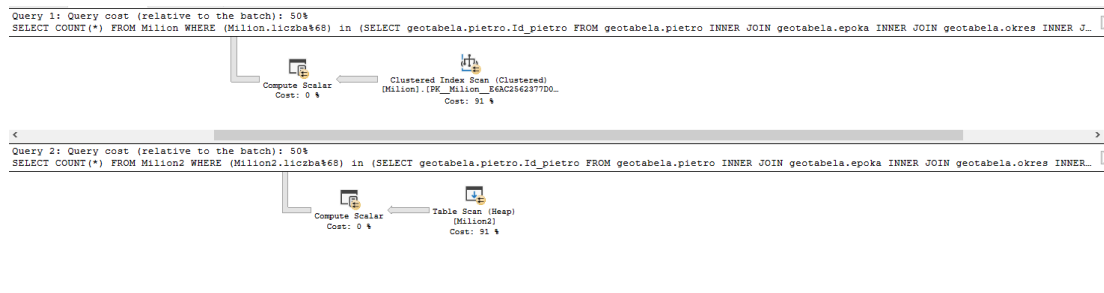
#### Dla Sql Server 2ZL



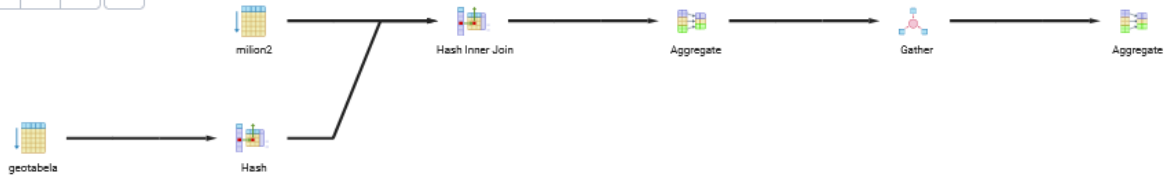
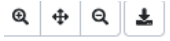
#### Dla Sql Server 3ZG



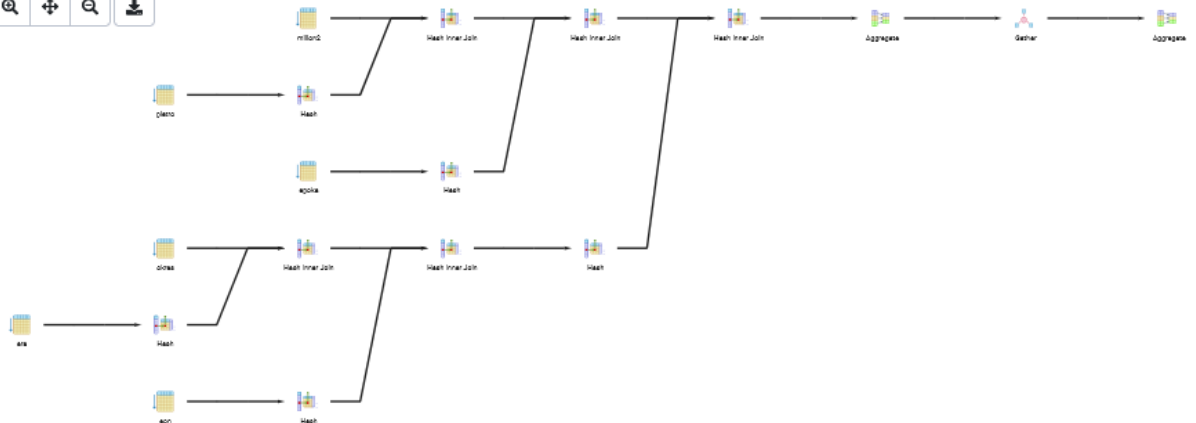
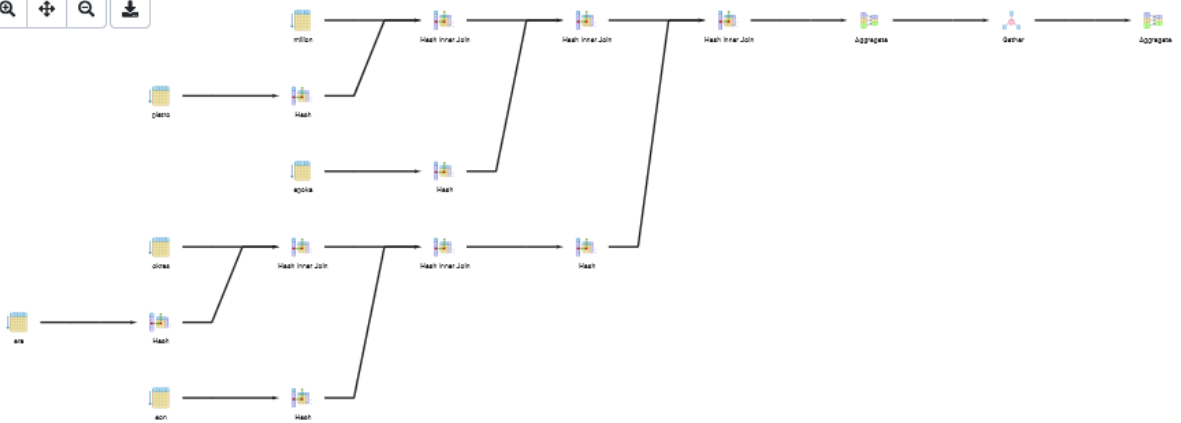
#### Dla Sql Server 4ZG



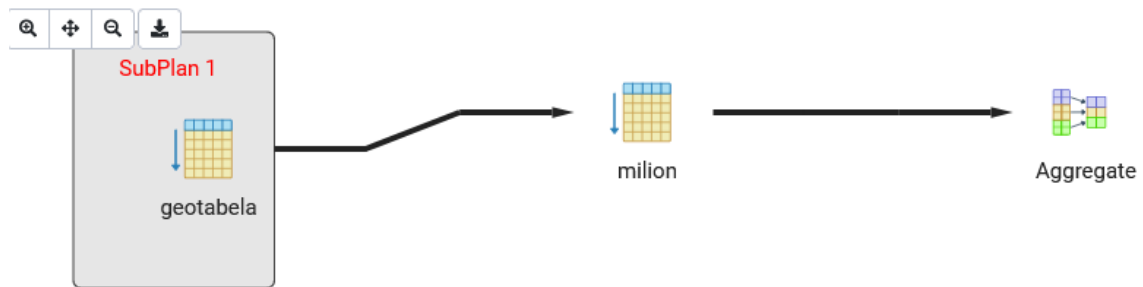
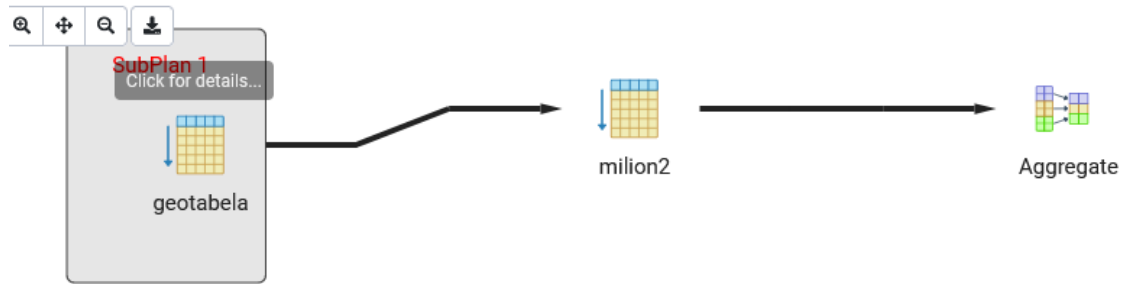
## Dla Postgres 1ZL



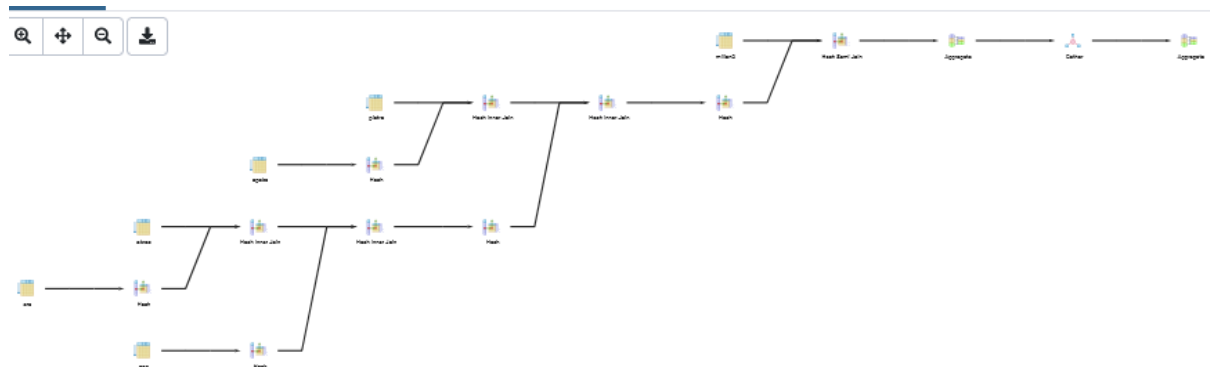
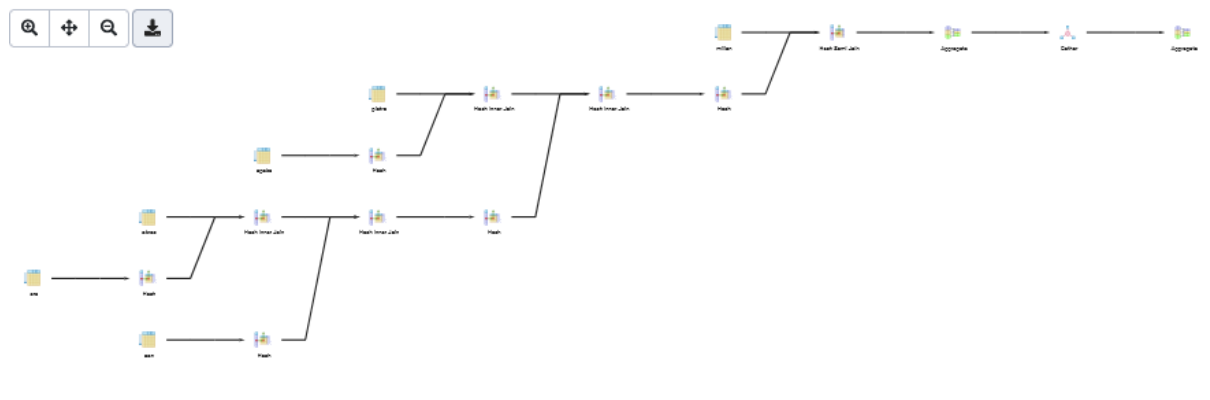
## Dla Postgres 2ZL



## Dla Postgres 3ZG



## Dla Postgres 4ZG



## **Wnioski:**

Zarówno dla oprogramowania Sql Server jak i Postgres zapytania dla tabel z indexami osiągnęły lepsze czasy.

Zaproponowane w artykule zapytania nie wykorzystują w pełni możliwości indexów, co pokazuje używanie przez algorytm "Scan Index", a nie "Seek Index". "Seek index" w tym przypadku nie jest możliwy do użycia ponieważ posegregowane dane (tabela z indexami) jest niszczone po użyciu funkcji modulo (wyniki nie będą posortowane), co powoduje, że zarówno tabela z indexami jak i bez jest nie jest posortowana dla tych zapytań i zmusza algorytm do przeszukania całej tabeli "Scan", zamiast dużo szybszej "Seek".