Zaawansowane technologie bazodanowe
Mikołaj Wielgos
Lab1

Indeksy oparte o haszowanie

```
explain analyze select * from zamowienia where idkompozycji = 'buk1';

Seq Scan on zamowienia  (cost=0.00..374.19 rows=424 width=68) (actual
time=0.205..0.990 rows=424 loops=1)
 Filter: (idkompozycji = 'buk1'::bpchar)
 Rows Removed by Filter: 7591
Planning Time: 0.294 ms
Execution Time: 1.027 ms
```

```
create index zamowienia_idkompozycji_idx on zamowienia using hash
(idkompozycji)
```

```
explain analyze select * from zamowienia where idkompozycji = 'buk1';

Bitmap Heap Scan on zamowienia  (cost=15.29..306.74 rows=424 width=68) (actual
time=0.031..0.141 rows=424 loops=1)
 Recheck Cond: (idkompozycji = 'buk1'::bpchar)
 Heap Blocks: exact=92
 ->  Bitmap Index Scan on zamowienia_idkompozycji_idx  (cost=0.00..15.18
rows=424 width=0) (actual time=0.019..0.019 rows=424 loops=1)
       Index Cond: (idkompozycji = 'buk1'::bpchar)
Planning Time: 0.101 ms
Execution Time: 0.178 ms
```

Indeksy oparte o b-drzewa

```
create index zamowienia_idkompozycji_idx on zamowienia (idkompozycji);
```

```
explain analyze select * from zamowienia where idkompozycji = 'buk1';

Bitmap Heap Scan on zamowienia  (cost=7.57..299.02 rows=424 width=68) (actual
time=0.027..0.147 rows=424 loops=1)
 Recheck Cond: (idkompozycji = 'buk1'::bpchar)
 Heap Blocks: exact=92
 ->  Bitmap Index Scan on zamowienia_idkompozycji_idx  (cost=0.00..7.46
rows=424 width=0) (actual time=0.018..0.018 rows=424 loops=1)
       Index Cond: (idkompozycji = 'buk1'::bpchar)
Planning Time: 0.167 ms
Execution Time: 0.177 ms
```

```
explain analyze select * from zamowienia where idkompozycji < 'c'

Bitmap Heap Scan on zamowienia  (cost=27.39..325.77 rows=1950 width=68) (actual
time=0.052..0.353 rows=1950 loops=1)
 Recheck Cond: (idkompozycji < 'c'::bpchar)
 Heap Blocks: exact=93
 ->  Bitmap Index Scan on zamowienia_idkompozycji_idx  (cost=0.00..26.91
rows=1950 width=0) (actual time=0.042..0.043 rows=1950 loops=1)
       Index Cond: (idkompozycji < 'c'::bpchar)
Planning Time: 0.074 ms
Execution Time: 0.429 ms
```

```
explain analyze select * from zamowienia where idkompozycji >= 'c';

Seq Scan on zamowienia  (cost=0.00..374.19 rows=6065 width=68) (actual
time=0.007..0.937 rows=6065 loops=1)
 Filter: (idkompozycji >= 'c'::bpchar)
 Rows Removed by Filter: 1950
Planning Time: 0.091 ms
Execution Time: 1.109 ms
```

```
SET enable_seqscan TO off;
```

```
explain analyze select * from zamowienia where idkompozycji < 'c';

Bitmap Heap Scan on zamowienia  (cost=27.39..325.77 rows=1950 width=68) (actual
time=0.058..0.410 rows=1950 loops=1)
 Recheck Cond: (idkompozycji < 'c'::bpchar)
 Heap Blocks: exact=93
 ->  Bitmap Index Scan on zamowienia_idkompozycji_idx  (cost=0.00..26.91
rows=1950 width=0) (actual time=0.047..0.047 rows=1950 loops=1)
       Index Cond: (idkompozycji < 'c'::bpchar)
Planning Time: 0.067 ms
Execution Time: 0.484 ms
```

```
explain analyze select * from zamowienia where idkompozycji >= 'c';

Bitmap Heap Scan on zamowienia  (cost=75.29..425.10 rows=6065 width=68) (actual
time=0.121..0.606 rows=6065 loops=1)
 Recheck Cond: (idkompozycji >= 'c'::bpchar)
 Heap Blocks: exact=94
 ->  Bitmap Index Scan on zamowienia_idkompozycji_idx  (cost=0.00..73.77
rows=6065 width=0) (actual time=0.110..0.110 rows=6065 loops=1)
       Index Cond: (idkompozycji >= 'c'::bpchar)
Planning Time: 0.061 ms
Execution Time: 0.785 ms
```

```
set enable_seqscan to on;
```

Indeksy a wzorce

```
create index zamowienia_uwagi_idx on zamowienia (uwagi);
```

```
explain analyse select * from zamowienia where uwagi like 'do%';

Seq Scan on zamowienia  (cost=0.00..374.19 rows=11 width=68) (actual
time=0.043..0.444 rows=11 loops=1)
 Filter: ((uwagi)::text ~~ 'do%'::text)
 Rows Removed by Filter: 8004
Planning Time: 0.226 ms
Execution Time: 0.459 ms
```

```
drop index zamowienia_uwagi_idx;
create index zamowienia_uwagi_idx on zamowienia (uwagi VARCHAR_PATTERN_OPS);
```

```
explain analyse select * from zamowienia where uwagi like 'do%';

Bitmap Heap Scan on zamowienia  (cost=4.40..41.92 rows=11 width=68) (actual
time=0.017..0.021 rows=11 loops=1)
 Filter: ((uwagi)::text ~~ 'do%'::text)
 Heap Blocks: exact=2
 ->  Bitmap Index Scan on zamowienia_uwagi_idx  (cost=0.00..4.39 rows=11
width=0) (actual time=0.010..0.011 rows=11 loops=1)
       Index Cond: (((uwagi)::text ~>=~ 'do'::text) AND ((uwagi)::text ~<~
'dp'::text))
Planning Time: 0.200 ms
Execution Time: 0.046 ms
```

Indeksy wielokolumnowe

```
create index zamowienia_idklienta_idodbiorcy_idkompozycji_idx on zamowienia
(idklienta, idodbiorcy, idkompozycji);
```

```
explain analyse select * from zamowienia where idklienta = 'msowins' and
idodbiorcy = 1 and idkompozycji = 'buk1';

Index Scan using zamowienia_idklienta_idodbiorcy_idkompozycji_idx on zamowienia
(cost=0.28..8.30 rows=1 width=68) (actual time=0.014..0.014 rows=1 loops=1)
 Index Cond: (((idklienta)::text = 'msowins'::text) AND (idodbiorcy = 1) AND
(idkompozycji = 'buk1'::bpchar))
Planning Time: 0.153 ms
Execution Time: 0.030 ms
```

```
explain analyse select * from zamowienia where idklienta = 'msowins' or
idodbiorcy = 1 or idkompozycji = 'buk1';

Seq Scan on zamowienia  (cost=0.00..414.26 rows=843 width=68) (actual
time=0.074..0.819 rows=841 loops=1)
 Filter: (((idklienta)::text = 'msowins'::text) OR (idodbiorcy = 1) OR
(idkompozycji = 'buk1'::bpchar))
 Rows Removed by Filter: 7174
Planning Time: 0.048 ms
Execution Time: 0.853 ms
```

```
explain analyse select * from zamowienia where idkompozycji = 'buk1';

Seq Scan on zamowienia  (cost=0.00..374.19 rows=424 width=68) (actual
time=0.091..0.669 rows=424 loops=1)
 Filter: (idkompozycji = 'buk1'::bpchar)
 Rows Removed by Filter: 7591
Planning Time: 0.062 ms
Execution Time: 0.694 ms
```

```
drop index zamowienia_idklienta_idodbiorcy_idkompozycji_idx;
create index zamowienia_idklienta_idx on zamowienia (idklienta);
create index zamowienia_idodbiorcy_idx on zamowienia (idodbiorcy);
create index zamowienia_idkompozycji_idx on zamowienia (idkompozycji);
```

```
explain analyse select * from zamowienia where idklienta = 'msowins' and
idodbiorcy = 1 and idkompozycji = 'buk1';

Bitmap Heap Scan on zamowienia  (cost=19.89..23.91 rows=1 width=68) (actual
time=0.051..0.052 rows=1 loops=1)
 Recheck Cond: (((idklienta)::text = 'msowins'::text) AND (idodbiorcy = 1) AND
(idkompozycji = 'buk1'::bpchar))
 Heap Blocks: exact=1
 ->  BitmapAnd  (cost=19.89..19.89 rows=1 width=0) (actual time=0.048..0.048
rows=0 loops=1)
       ->  Bitmap Index Scan on zamowienia_idklienta_idx  (cost=0.00..5.49
rows=161 width=0) (actual time=0.017..0.017 rows=161 loops=1)
             Index Cond: ((idklienta)::text = 'msowins'::text)
       ->  Bitmap Index Scan on zamowienia_idodbiorcy_idx  (cost=0.00..6.43
rows=287 width=0) (actual time=0.013..0.013 rows=287 loops=1)
             Index Cond: (idodbiorcy = 1)
       ->  Bitmap Index Scan on zamowienia_idkompozycji_idx  (cost=0.00..7.46
rows=424 width=0) (actual time=0.015..0.015 rows=424 loops=1)
             Index Cond: (idkompozycji = 'buk1'::bpchar)
```

```
explain analyse select * from zamowienia where idklienta = 'msowins' or
idodbiorcy = 1 or idkompozycji = 'buk1';

Bitmap Heap Scan on zamowienia  (cost=20.02..309.28 rows=843 width=68) (actual
time=0.055..0.263 rows=841 loops=1)
 Recheck Cond: (((idklienta)::text = 'msowins'::text) OR (idodbiorcy = 1) OR
(idkompozycji = 'buk1'::bpchar))
 Heap Blocks: exact=93
 ->  BitmapOr  (cost=20.02..20.02 rows=872 width=0) (actual time=0.044..0.044
rows=0 loops=1)
       ->  Bitmap Index Scan on zamowienia_idklienta_idx  (cost=0.00..5.49
rows=161 width=0) (actual time=0.024..0.024 rows=161 loops=1)
             Index Cond: ((idklienta)::text = 'msowins'::text)
       ->  Bitmap Index Scan on zamowienia_idodbiorcy_idx  (cost=0.00..6.43
rows=287 width=0) (actual time=0.008..0.008 rows=287 loops=1)
             Index Cond: (idodbiorcy = 1)
       ->  Bitmap Index Scan on zamowienia_idkompozycji_idx  (cost=0.00..7.46
rows=424 width=0) (actual time=0.011..0.011 rows=424 loops=1)
             Index Cond: (idkompozycji = 'buk1'::bpchar)
```

```
explain analyse select * from zamowienia where idkompozycji = 'buk1';

Bitmap Heap Scan on zamowienia  (cost=7.57..299.02 rows=424 width=68) (actual
time=0.032..0.168 rows=424 loops=1)
 Recheck Cond: (idkompozycji = 'buk1'::bpchar)
 Heap Blocks: exact=92
 ->  Bitmap Index Scan on zamowienia_idkompozycji_idx  (cost=0.00..7.46
rows=424 width=0) (actual time=0.023..0.023 rows=424 loops=1)
       Index Cond: (idkompozycji = 'buk1'::bpchar)
Planning Time: 0.065 ms
Execution Time: 0.196 ms
```

Indeksy a sortowanie

```
explain analyse select * from zamowienia order by idkompozycji;

Sort  (cost=873.86..893.90 rows=8015 width=68) (actual time=2.527..2.787
rows=8015 loops=1)
 Sort Key: idkompozycji
 Sort Method: quicksort  Memory: 881kB
 ->  Seq Scan on zamowienia  (cost=0.00..354.15 rows=8015 width=68) (actual
time=0.004..0.531 rows=8015 loops=1)
Planning Time: 0.081 ms
Execution Time: 3.039 ms
```

```
set enable_seqscan to off;
explain analyse select * from zamowienia order by idkompozycji;

Index Scan using zamowienia_idkompozycji_idx on zamowienia  (cost=0.28..1250.49
rows=8015 width=68) (actual time=0.026..1.675 rows=8015 loops=1)
Planning Time: 0.068 ms
Execution Time: 1.914 ms
```

```
drop index zamowienia_idkompozycji_idx;
```

```
explain analyse select * from zamowienia order by idkompozycji;

Sort  (cost=873.86..893.90 rows=8015 width=68) (actual time=2.208..2.468
rows=8015 loops=1)
 Sort Key: idkompozycji
 Sort Method: quicksort  Memory: 881kB
 ->  Seq Scan on zamowienia  (cost=0.00..354.15 rows=8015 width=68) (actual
time=0.003..0.507 rows=8015 loops=1)
Planning Time: 0.080 ms
Execution Time: 2.774 ms
```

Indeksy częściowe

```
create index zamowienia_idklienta_idx on zamowienia (idklienta);
```

```
explain analyse select * from zamowienia where idklienta ='gurbanik' and
zaplacone = true;

Bitmap Heap Scan on zamowienia  (cost=5.45..248.07 rows=151 width=68) (actual
time=0.041..0.145 rows=151 loops=1)
 Recheck Cond: (((idklienta)::text = 'gurbanik'::text) AND zaplacone)
 Heap Blocks: exact=72
 ->  Bitmap Index Scan on zamowienia_idklienta_idx  (cost=0.00..5.42 rows=151
width=0) (actual time=0.033..0.033 rows=151 loops=1)
       Index Cond: ((idklienta)::text = 'gurbanik'::text)
Planning Time: 0.199 ms
Execution Time: 0.169 ms
```

```
explain analyse select * from zamowienia where idklienta ='gurbanik' and
zaplacone = false;

Seq Scan on zamowienia  (cost=0.00..374.19 rows=1 width=68) (actual
time=0.574..0.574 rows=0 loops=1)
 Filter: ((NOT zaplacone) AND ((idklienta)::text = 'gurbanik'::text))
 Rows Removed by Filter: 8015
Planning Time: 0.074 ms
Execution Time: 0.592 ms
```

```
explain analyse select sum(cena) from zamowienia where zaplacone = false;

Aggregate  (cost=354.17..354.18 rows=1 width=32) (actual time=0.592..0.593
rows=1 loops=1)
 ->  Seq Scan on zamowienia  (cost=0.00..354.15 rows=7 width=5) (actual
time=0.084..0.585 rows=7 loops=1)
       Filter: (NOT zaplacone)
       Rows Removed by Filter: 8008
Planning Time: 0.066 ms
Execution Time: 0.617 ms
```

Indeksy na wyrażeniach

```
create index klienci_miasto_idx on klienci(lower(miasto) VARCHAR_PATTERN_OPS);
```

```
set enable_seqscan to off;
```

```
explain analyse select * from klienci where lower(miasto) like 'krak%';

Index Scan using klienci_miasto_idx on klienci  (cost=0.14..8.17 rows=1
width=692) (actual time=0.014..0.022 rows=23 loops=1)
 Index Cond: ((lower((miasto)::text) ~>=~ 'krak'::text) AND
(lower((miasto)::text) ~<~ 'kral'::text))
 Filter: (lower((miasto)::text) ~~ 'krak%'::text)
Planning Time: 0.181 ms
Execution Time: 0.071 ms
```

```
set enable_seqscan to on;
```

```
explain analyse select * from klienci where lower(miasto) like 'krak%';

Seq Scan on klienci  (cost=0.00..1.75 rows=1 width=692) (actual
time=0.015..0.025 rows=23 loops=1)
 Filter: (lower((miasto)::text) ~~ 'krak%'::text)
 Rows Removed by Filter: 27
Planning Time: 0.088 ms
Execution Time: 0.042 ms
```

Indeksy GiST

```
explain select * from zamowienia where point '(50,50)' <-> lokalizacja <= 10;

Seq Scan on zamowienia  (cost=0.00..394.23 rows=2672 width=68)
"  Filter: (('(50,50)'::point <-> lokalizacja) <= '10'::double precision)"
```

```
explain analyze select * from zamowienia where point '(50,50)' >> lokalizacja
and point '(50,50)' <<| lokalizacja;

Seq Scan on zamowienia  (cost=0.00..394.23 rows=80 width=68) (actual
time=0.082..0.769 rows=1979 loops=1)
"  Filter: (('(50,50)'::point >> lokalizacja) AND ('(50,50)'::point <<|
lokalizacja))"
 Rows Removed by Filter: 6036
Planning Time: 0.067 ms
Execution Time: 0.840 ms
```

```
create index zamowienia_lokalizacja_idx on zamowienia using gist (lokalizacja);
```

```
VACUUM ANALYZE zamowienia;
set enable_seqscan to off;
```

```
explain analyze select * from zamowienia where point '(50,50)' <-> lokalizacja
<= 10;

Seq Scan on zamowienia  (cost=10000000000.00..10000000394.23 rows=2672
width=68) (actual time=26.678..27.248 rows=250 loops=1)
"  Filter: (('(50,50)'::point <-> lokalizacja) <= '10'::double precision)"
 Rows Removed by Filter: 7765
Planning Time: 0.072 ms
JIT:
 Functions: 2
"  Options: Inlining true, Optimization true, Expressions true, Deforming true"
"  Timing: Generation 0.383 ms, Inlining 2.445 ms, Optimization 14.511 ms,
Emission 9.605 ms, Total 26.943 ms"
Execution Time: 27.677 ms
```

Nie udało mi się zmusić wykorzystania indeksu. Problemu szukałem w:
braku PostGIS, lecz był włączony w bazie:

```
SELECT PostGIS_version();
3.4 USE_GEOS=1 USE_PROJ=1 USE_STATS=1
```

Użycia funkcji zamiast operatorów, jednak spotkałem się z błędem.

```
explain analyze select * from zamowienia where st_distance(st_makepoint(50,
50), lokalizacja) <= 10
```

[2024-10-25 06:32:24] [42883] BŁĄD: funkcja st_distance(geometry, point) nie istnieje
[2024-10-25 06:32:24] Hint: Brak funkcji pasującej do podanej nazwy i typów argumentów. Być może należy dodać jawne rzutowanie typów.
[2024-10-25 06:32:24] Position: 48