

# Sprawozdanie - Zaawansowane technologie bazodanowe

Laboratorium 5 - 20.12.2019

## Indeksy

Dominik Wróbel

### Zadanie 1 – indeksy oparte o haszowanie

```
1. SELECT * FROM zamowienia WHERE idkompozycji = 'buk1';
2. EXPLAIN ANALYZE VERBOSE SELECT * FROM zamowienia WHERE idkompozycji = 'buk1';
3.
4.
5.
6. wrobdom1=> EXPLAIN ANALYZE VERBOSE SELECT * FROM zamowienia WHERE idkompozycji = 'buk1';
7. QUERY PLAN
8.
9. Seq Scan on public.zamowienia (cost=0.00..167.19 rows=424 width=52) (actual time=0.023..2.63
10. 8 rows=424 loops=1)
11.   Output: idzamowienia, idklienta, idodbiorcy, idkompozycji, termin, cena, zaplacone, uwagi
12.   Filter: (zamowienia.idkompozycji = 'buk1'::bpchar)
13.   Rows Removed by Filter: 7591
14.   Planning time: 0.114 ms
15.   Execution time: 2.710 ms
16. (6 rows)
17. CREATE INDEX indIdKom ON zamowienia USING hash (idkompozycji);
18. CREATE INDEX
19. wrobdom1=> EXPLAIN ANALYZE VERBOSE SELECT * FROM zamowienia WHERE idkompozycji = 'buk1';
20. QUERY PLAN
21.
22. Bitmap Heap Scan on public.zamowienia (cost=15.29..87.59 rows=424 width=52) (actual time=0.1
23. 37..0.550 rows=424 loops=1)
24.   Output: idzamowienia, idklienta, idodbiorcy, idkompozycji, termin, cena, zaplacone, uwagi
25.   Recheck Cond: (zamowienia.idkompozycji = 'buk1'::bpchar)
26.   Heap Blocks: exact=67
27.   > Bitmap Index Scan on indidkom (cost=0.00..15.18 rows=424 width=0) (actual time=0.081..0.08
28.   1 rows=424 loops=1)
29.     Index Cond: (zamowienia.idkompozycji = 'buk1'::bpchar)
30.   Planning time: 0.257 ms
31.   Execution time: 0.635 ms
32. (8 rows)
```

### Zadanie 2 – indeksy oparte o b-drzewa

```
1. wrobdom1=> DROP INDEX indIdKom;
2. DROP INDEX
3.
4. wrobdom1=> CREATE INDEX indIdKom ON zamowienia (idkompozycji);
5. CREATE INDEX
6. wrobdom1=> EXPLAIN ANALYZE VERBOSE SELECT * FROM zamowienia WHERE idkompozycji = 'buk1';
7.
```

```

8. QUERY PLAN
9. -----
10. Bitmap Heap Scan on public.zamowienia (cost=11.57..83.87 rows=424 width=52) (actual
11.   time=0.273..0.606 rows=424 loops=1)
12.   Output: idzamowienia, idklienta, idodbiorcy, idkompozycji, termin, cena, zaplacon
13.   e, uwagi
14.   Recheck Cond: (zamowienia.idkompozycji = 'buk1'::bpchar)
15.   Heap Blocks: exact=67
16.   -
17.   > Bitmap Index Scan on indidkom (cost=0.00..11.46 rows=424 width=0) (actual time=0
18.     .246..0.246 rows=424 loops=1)
19.     Index Cond: (zamowienia.idkompozycji = 'buk1'::bpchar)
20. Planning time: 0.365 ms
21. Execution time: 0.692 ms
22. (8 rows)

```

### Zadanie 3 – badanie stosowania indeksu

Indeks nie jest wykorzystywany.

```

1. EXPLAIN ANALYZE VERBOSE SELECT * FROM zamowienia WHERE idkompozycji < 'b%';
2.
3. QUERY PLAN
4. -----
5. Index Scan using indidkom on public.zamowienia (cost=0.28..8.30 rows=1 width=52) (
6.   actual time=0.023..0.024 rows=0 loops=1)
7.   Output: idzamowienia, idklienta, idodbiorcy, idkompozycji, termin, cena, zaplacon
8.   e, uwagi
9.   Index Cond: (zamowienia.idkompozycji < '%b'::bpchar)
10. Planning time: 0.143 ms
11. Execution time: 0.060 ms
12. (5 rows)
13.
14. wrobdom1=> EXPLAIN ANALYZE VERBOSE SELECT * FROM zamowienia WHERE idkompozycji > 'b%
15.   ';
16. QUERY PLAN
17. -----
18. Seq Scan on public.zamowienia (cost=0.00..167.19 rows=8015 width=52) (actual time=
19.   0.027..4.570 rows=8015 loops=1)
20.   Output: idzamowienia, idklienta, idodbiorcy, idkompozycji, termin, cena, zaplacon
21.   e, uwagi
22.   Filter: (zamowienia.idkompozycji > '%b'::bpchar)
23. Planning time: 0.152 ms
24. Execution time: 5.271 ms
25. (5 rows)

```

Wymuszenie stosowanie indeksu:

```

1. wrobdom1=> SET ENABLE_SEQSCAN TO OFF;
2. SET
3. wrobdom1=> EXPLAIN ANALYZE VERBOSE SELECT * FROM zamowienia WHERE idkompozycji > 'b%
4.   ';
5. QUERY PLAN

```

```

6. -----
7. Bitmap Heap Scan on public.zamowienia (cost=158.40..325.59 rows=8015 width=52) (ac
   tual time=2.594..4.714 rows=8015 loops=1)
8.   Output: idzamowienia, idklienta, idodbiorcy, idkompozycji, termin, cena, zaplacon
   e, uwagi
9.   Recheck Cond: (zamowienia.idkompozycji > '%b'::bpchar)
10.  Heap Blocks: exact=67
11.  -
   > Bitmap Index Scan on indidkom (cost=0.00..156.40 rows=8015 width=0) (actual time
   =2.554..2.554 rows=8015 loops=1)
12.   Index Cond: (zamowienia.idkompozycji > '%b'::bpchar)
13. Planning time: 0.193 ms
14. Execution time: 5.444 ms

```

## Zadanie 4 - Indeksy a wzorce

Indeks nie jest wykorzystywany:

```

1. wrobdom1=> DROP INDEX indIdKom;
2. DROP INDEX
3.
4. wrobdom1=> CREATE INDEX indUwagi ON zamowienia (uwagi);
5. CREATE INDEX
6.
7. wrobdom1=> SET ENABLE_SEQSCAN TO ON;
8. SET
9.
10. wrobdom1=> EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE uwagi LIKE 'do%';
11.
12. QUERY PLAN
13. -----
14. Seq Scan on zamowienia (cost=0.00..167.19 rows=10 width=52) (actual time=0.019..2.
   448 rows=11 loops=1)
15.   Filter: ((uwagi)::text ~~ 'do% '::text)
16.   Rows Removed by Filter: 8004
17. Planning time: 0.296 ms
18. Execution time: 2.485 ms
19. (5 rows)

```

Indeks jest wykorzystywany:

```

1. wrobdom1=> DROP INDEX indUwagi;
2. DROP INDEX
3.
4.
5. wrobdom1=> CREATE INDEX indUwagi ON zamowienia (uwagi varchar_pattern_ops);
6. CREATE INDEX
7. wrobdom1=> EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE uwagi LIKE 'do%';
8.
9.
10. QUERY PLAN
11. -----
12. Bitmap Heap Scan on zamowienia (cost=4.40..35.16 rows=11 width=52) (actual time=0.
   034..0.037 rows=11 loops=1)
13.   Filter: ((uwagi)::text ~~ 'do% '::text)
14.   Heap Blocks: exact=1

```

```

15. -
    > Bitmap Index Scan on induwagi (cost=0.00..4.39 rows=11 width=0) (actual time=0.0
    23..0.023 rows=11 loops=1)
16.      Index Cond: (((uwagi)::text ~>~ 'do'::text) AND ((uwagi)::text ~<~ 'dp'::t
    ext))
17. Planning time: 0.452 ms
18. Execution time: 0.075 ms
19. (7 rows)
20.
21.
22. wrobdom1=> DROP INDEX indUwagi;
23. DROP INDEX

```

## Zadanie 5 – Indeksy wielokolumnowe

Indeks dla wielu kolumn:

```

1. wrobdom1=> CREATE INDEX multi on zamowienia (idklienta, idodbiorcy, idkompozycji);
2. CREATE INDEX
3.
4. wrobdom1=> EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE idklienta = 'mrusinek' AND
    idodbiorcy = 23 AND idkompozycji = 'ko4';
5.
6.                                QUERY PLAN
7. -----
8.      Index Scan using multi on zamowienia (cost=0.28..8.30 rows=1 width=52) (actual tim
    e=0.042..0.044 rows=1 loops=1)
9.        Index Cond: (((idklienta)::text = 'mrusinek'::text) AND (idodbiorcy = 23) AND (id
    kompozycji = 'ko4'::bpchar))
10. Planning time: 0.193 ms
11. Execution time: 0.090 ms
12. (4 rows)
13.
14.
15. wrobdom1=> EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE idklienta = 'mrusinek' OR
    idodbiorcy = 23 OR idkompozycji = 'ko4';
16.
17.                                QUERY PLAN
18. -----
19.      Seq Scan on zamowienia (cost=0.00..207.26 rows=850 width=52) (actual time=0.028..3
    .274 rows=846 loops=1)
20.        Filter: (((idklienta)::text = 'mrusinek'::text) OR (idodbiorcy = 23) OR (idkompoz
    ycji = 'ko4'::bpchar))
21.        Rows Removed by Filter: 7169
22. Planning time: 0.227 ms
23. Execution time: 3.390 ms
24. (5 rows)
25.
26.
27. wrobdom1=> EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE idkompozycji = 'buk1';
28.
29.                                QUERY PLAN
30. -----
31.      Seq Scan on zamowienia (cost=0.00..167.19 rows=424 width=52) (actual time=0.021..2
    .588 rows=424 loops=1)
32.        Filter: (idkompozycji = 'buk1'::bpchar)

```

```

33. Rows Removed by Filter: 7591
34. Planning time: 0.146 ms
35. Execution time: 2.659 ms
36. (5 rows)
37.
38. wrobdom1=> DROP INDEX multi;
39. DROP INDEX

```

Indeks dla każdej kolumny z osobna:

```

1. wrobdom1=> CREATE INDEX indIdKlient on zamowienia(idklienta);
2. CREATE INDEX
3. wrobdom1=> CREATE INDEX indIdOdbiorcy on zamowienia(idodbiorcy);
4. CREATE INDEX
5. wrobdom1=> CREATE INDEX indIdKompozycji on zamowienia(idkompozycji);
6. CREATE INDEX
7.
8. EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE idklienta = 'mrusinek' AND idodbiorcy
  = 23 AND idkompozycji = 'ko4';
9.
10. QUERY PLAN
11. -----
12. Bitmap Heap Scan on zamowienia (cost=23.95..27.97 rows=1 width=52) (actual time=0.
  236..0.237 rows=1 loops=1)
13.   Recheck Cond: (((idklienta)::text = 'mrusinek'::text) AND (idodbiorcy = 23) AND (
  idkompozycji = 'ko4'::bpchar))
14.   Heap Blocks: exact=1
15.   -
  > BitmapAnd (cost=23.95..23.95 rows=1 width=0) (actual time=0.229..0.229 rows=0 lo
  ops=1)
16.   -
  > Bitmap Index Scan on indidklient (cost=0.00..5.48 rows=160 width=0) (actual time
  =0.073..0.073 rows=160 loops=1)
17.     Index Cond: ((idklienta)::text = 'mrusinek'::text)
18.   -
  > Bitmap Index Scan on indidodbiorcy (cost=0.00..6.61 rows=311 width=0) (actual ti
  me=0.065..0.065 rows=311 loops=1)
19.     Index Cond: (idodbiorcy = 23)
20.   -
  > Bitmap Index Scan on indidkompozycji (cost=0.00..11.35 rows=409 width=0) (actual
  time=0.081..0.081 rows=409 loops=1)
21.     Index Cond: (idkompozycji = 'ko4'::bpchar)
22. Planning time: 6.633 ms
23. Execution time: 0.289 ms
24. (12 rows)
25.
26.
27. EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE idklienta = 'mrusinek' OR idodbiorcy
  = 23 OR idkompozycji = 'ko4';
28.
29. QUERY PLAN
30. -----
31. Bitmap Heap Scan on zamowienia (cost=24.09..106.49 rows=850 width=52) (actual time
  =0.193..0.671 rows=846 loops=1)
32.   Recheck Cond: (((idklienta)::text = 'mrusinek'::text) OR (idodbiorcy = 23) OR (id
  kompozycji = 'ko4'::bpchar))
33.   Heap Blocks: exact=67
34.   -
  > BitmapOr (cost=24.09..24.09 rows=880 width=0) (actual time=0.165..0.165 rows=0 l
  oops=1)

```

```

35.      -
      > Bitmap Index Scan on indidklient (cost=0.00..5.48 rows=160 width=0) (actual time
      =0.053..0.053 rows=160 loops=1)
36.          Index Cond: ((idklienta)::text = 'mrusinek'::text)
37.      -
      > Bitmap Index Scan on indidodbiorcy (cost=0.00..6.61 rows=311 width=0) (actual ti
      me=0.041..0.042 rows=311 loops=1)
38.          Index Cond: (idodbiorcy = 23)
39.      -
      > Bitmap Index Scan on indidkompozycji (cost=0.00..11.35 rows=409 width=0) (actual
      time=0.069..0.069 rows=409 loops=1)
40.          Index Cond: (idkompozycji = 'ko4'::bpchar)
41. Planning time: 0.188 ms
42. Execution time: 0.800 ms
43. (12 rows)
44.
45.
46. EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE idkompozycji = 'buk1';
47.
48.                                     QUERY PLAN
49. -----
50. Bitmap Heap Scan on zamowienia (cost=11.57..83.87 rows=424 width=52) (actual time=
    0.183..0.512 rows=424 loops=1)
51.   Recheck Cond: (idkompozycji = 'buk1'::bpchar)
52.   Heap Blocks: exact=67
53.   -
      > Bitmap Index Scan on indidkompozycji (cost=0.00..11.46 rows=424 width=0) (actual
      time=0.157..0.157 rows=424 loops=1)
54.       Index Cond: (idkompozycji = 'buk1'::bpchar)
55. Planning time: 0.155 ms
56. Execution time: 0.594 ms
57. (7 rows)

```

## Zadanie 6 – Indeksy a sortowanie

```

1. EXPLAIN ANALYZE VERBOSE SELECT * FROM zamowienia ORDER BY idkompozycji;
2.
3.                                     QUERY PLAN
4. -----
5. Index Scan using indidkompozycji on public.zamowienia (cost=0.28..484.02 rows=8015
   width=52) (actual time=0.019..4.842 rows=8015 loops=1)
6.   Output: idzamowienia, idklienta, idodbiorcy, idkompozycji, termin, cena, zaplacon
   e, uwagi
7. Planning time: 0.185 ms
8. Execution time: 5.590 ms
9. (4 rows)
10.
11.
12. wrobdom1=> DROP INDEX indIdKompozycji;
13. DROP INDEX
14.
15.
16.                                     QUERY PLAN

```

```

17. -----
18. Sort (cost=666.86..686.90 rows=8015 width=52) (actual time=11.117..11.807 rows=8015 loops=1)
19.   Output: idzamowienia, idklienta, idodbiorcy, idkompozycji, termin, cena, zaplacone, uwagi
20.   Sort Key: zamowienia.idkompozycji
21.   Sort Method: quicksort Memory: 819kB
22.   -
23.   > Seq Scan on public.zamowienia (cost=0.00..147.15 rows=8015 width=52) (actual time=0.016..1.777 rows=8015 loops=1)
24.     Output: idzamowienia, idklienta, idodbiorcy, idkompozycji, termin, cena, zaplacone, uwagi
25.     Planning time: 0.174 ms
26.     Execution time: 12.275 ms
27. (8 rows)
28.
29.
30. wrobdom1=> DROP INDEX indIdOdbiorcy;
31. DROP INDEX
32. wrobdom1=> DROP INDEX indIdKlient;
33. DROP INDEX

```

## Zadanie 7 – Indeksy częściowe

W tych przypadkach indeks używany jest tylko tam gdzie pytamy o zapłacone zamówienia.

```

1. wrobdom1=> CREATE INDEX indIdKlient ON zamowienia(idklienta) WHERE zaplacone;
2. CREATE INDEX
3.
4. EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE idklienta='msowins' AND zaplacone;
5.
6. QUERY PLAN
7. -----
8. Bitmap Heap Scan on zamowienia (cost=5.53..74.54 rows=161 width=52) (actual time=0.099..0.289 rows=160 loops=1)
9.   Recheck Cond: (((idklienta)::text = 'msowins'::text) AND zaplacone)
10.   Heap Blocks: exact=63
11.   -
12.   > Bitmap Index Scan on indidklient (cost=0.00..5.49 rows=161 width=0) (actual time=0.076..0.076 rows=160 loops=1)
13.     Index Cond: ((idklienta)::text = 'msowins'::text)
14.   Planning time: 0.367 ms
15.   Execution time: 0.348 ms
16. (7 rows)
17. EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE idklienta='msowins' AND NOT zaplacone;
18.
19. QUERY PLAN
20. -----
21. Seq Scan on zamowienia (cost=0.00..167.19 rows=1 width=52) (actual time=0.021..2.444 rows=1 loops=1)
22.   Filter: ((NOT zaplacone) AND ((idklienta)::text = 'msowins'::text))
23.   Rows Removed by Filter: 8014
24.   Planning time: 0.128 ms

```

```

25. Execution time: 2.477 ms
26. (5 rows)
27.
28. EXPLAIN ANALYZE SELECT sum(cena) FROM zamowienia WHERE NOT zaplacone;
29.
30.                                QUERY PLAN
31. -----
32. Aggregate  (cost=147.17..147.18 rows=1 width=32) (actual time=1.480..1.480 rows=1 loops=1)
33.   -
34.   > Seq Scan on zamowienia  (cost=0.00..147.15 rows=7 width=5) (actual time=0.015..1.460 rows=7 loops=1)
35.     Filter: (NOT zaplacone)
36.     Rows Removed by Filter: 8008
37. Planning time: 0.490 ms
38. Execution time: 1.543 ms
39. (6 rows)

```

## Zadanie 8 – Indeksy na wyrażeniach

```

1. wrobdom1=> CREATE INDEX indexMiasto ON klienci (lower(miasto) varchar_pattern_ops);
2. CREATE INDEX
3.
4.
5.
6. EXPLAIN ANALYZE SELECT * FROM klienci WHERE lower(miasto) LIKE 'krak%';
7.
8. wrobdom1=> EXPLAIN ANALYZE SELECT * FROM klienci WHERE lower(miasto) LIKE 'krak%';
9.                                QUERY PLAN
10. -----
11. Seq Scan on klienci  (cost=0.00..1.75 rows=1 width=692) (actual time=0.028..0.084 rows=23 loops=1)
12.   Filter: (lower((miasto)::text) ~ 'krak% '::text)
13.   Rows Removed by Filter: 27
14. Planning time: 0.561 ms
15. Execution time: 0.122 ms
16. (5 rows)

```

## Zadanie 9 – Indeksy GiST

```

1. wrobdom1=> ALTER TABLE zamowienia ADD COLUMN lokalizacja point;
2. ALTER TABLE
3. wrobdom1=> UPDATE zamowienia SET lokalizacja=point(random()*100, random()*100);
4. UPDATE 8015
5.
6.
7.
8.

```



```

9. EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE lokalizacja <-> point(50,50) <= 10;
10.
11. QUERY PLAN
12. -----
13. Seq Scan on zamowienia (cost=0.00..278.23 rows=2672 width=68) (actual time=0.123..
    3.043 rows=242 loops=1)
14.   Filter: ((lokalizacja <-> '(50,50)::point) <= '10'::double precision)
15.   Rows Removed by Filter: 7773
16. Planning time: 0.366 ms
17. Execution time: 3.095 ms
18. (5 rows)
19.
20.
21. EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE box'((0,0), (50,50))' @> lokalizacja;
22.
23. QUERY PLAN
24. -----
25. Seq Scan on zamowienia (cost=0.00..258.19 rows=8 width=68) (actual time=0.107..2.7
    76 rows=2050 loops=1)
26.   Filter: ('(50,50),(0,0)::box @> lokalizacja)
27.   Rows Removed by Filter: 5965
28. Planning time: 0.089 ms
29. Execution time: 2.993 ms
30. (5 rows)
31.
32.
33. wrobdom1=> CREATE INDEX indexLok ON zamowienia USING GIST (lokalizacja);
34. CREATE INDEX
35.
36. wrobdom1=> EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE lokalizacja <-
    > point(50,50) <= 10;
37.
38. QUERY PLAN
39. -----
40. Seq Scan on zamowienia (cost=1000000000.00..10000000278.23 rows=2672 width=68) (a
    ctual time=0.128..3.105 rows=242 loops=1)
41.   Filter: ((lokalizacja <-> '(50,50)::point) <= '10'::double precision)
42.   Rows Removed by Filter: 7773
43. Planning time: 0.141 ms
44. Execution time: 3.157 ms
45. (5 rows)
46.
47. wrobdom1=> EXPLAIN ANALYZE SELECT * FROM zamowienia WHERE box'((0,0), (50,50))' @> l
    okalizacja;
48.
49.
50. QUERY PLAN
51. -----
52. Bitmap Heap Scan on zamowienia (cost=4.21..30.91 rows=8 width=68) (actual time=0.4
    89..1.434 rows=2050 loops=1)
53.   Recheck Cond: ('(50,50),(0,0)::box @> lokalizacja)
54.   Heap Blocks: exact=92
55.   -
    > Bitmap Index Scan on indexlok (cost=0.00..4.21 rows=8 width=0) (actual time=0.45
    9..0.459 rows=2050 loops=1)
56.       Index Cond: ('(50,50),(0,0)::box @> lokalizacja)
57. Planning time: 0.184 ms

```

58. Execution time: 1.665 ms  
59. (7 rows)