

Text

# Indeksowanie w bazach danych

Sebastian Ernst

Katedra Informatyki Stosowanej AGH

5 grudnia 2013

# Outline

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

**1** Indeksy

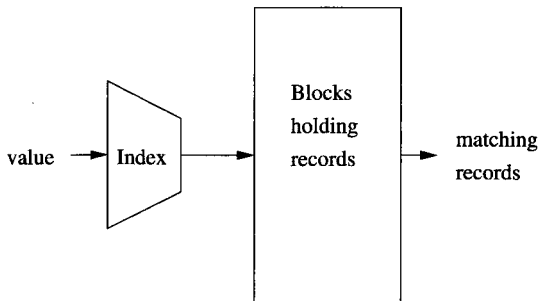
**2** Indeksy wielokolumnowe

**3** Indeksy uporządkowane

**4** Zagadnienia zaawansowane

# Czym jest indeks?

Indeks to struktura, która ma przyspieszyć wyszukiwanie danych. Indeks definiowany jest dla atrybutów, które nazywamy *kluczami indeksu* lub *kluczami wyszukiwania*.



Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

# Indeksy gęste i rzadkie

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

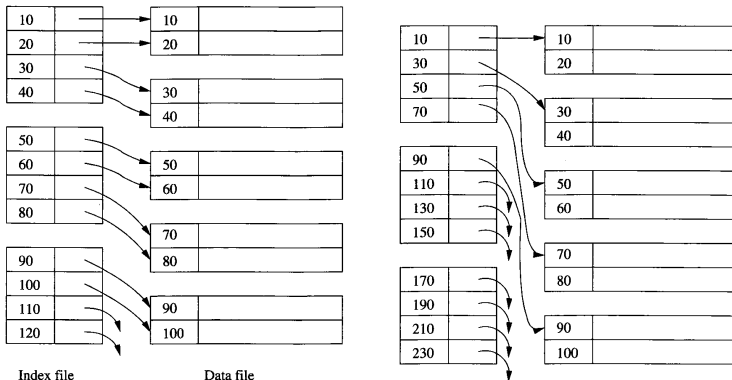
Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

- Indeks *gęsty* to taki, który posiada wpis dla każdej wartości klucza wyszukiwania.
- Indeks *rzadki* posiada wpisy tylko dla niektórych wartości.



# Przykład

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Mamy tabelę:

```
CREATE TABLE test1 (  
    id integer,  
    content varchar  
);
```

Załóżmy że aplikacja wykonuje wiele zapytań typu:

```
SELECT content FROM test1 WHERE id = constant;
```

Silnik bazodanowy za każdym razem musiałby przejrzeć wszystkie rekordy.

Indeks pozwala na przyspieszenie operacji:

```
CREATE INDEX test1_id_index ON test1 (id);
```

# Typy indeksów w PostgreSQL

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

PostgreSQL 9.3 wspiera następujące typy indeksów:

- B-drzewa (B-tree),
- z haszowaniem (hash),
- GiST,
- SP-GiST,
- GIN.

# Indeksy oparte o B-drzewa

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Indeksy oparte o B-drzewa mogą być stosowane dla dziedzin dla których określono relację porządkującą. Mogą więc zostać użyte jeżeli wartość klucza wyszukiwania porównywana jest przy pomocy następujących operatorów:

<

<=

=

>=

>

Zastosowanie mają też operatory BETWEEN oraz IN a także funkcje sprawdzające NULL.

# Indeksy oparte o B-drzewa, c.d.

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Porównania wykorzystujące dopasowanie wzorców mogą korzystać z indeksów opartych o B-drzewa **tylko** jeżeli wzorzec jest **stałą** i jest zaczepiony na początku ciągu znaków: `col LIKE 'foo%'`, `col ~ '^foo'`.

Porównania nie rozróżniające wielkości liter (`ILIKE`, `~*`) użyją indeksu, jeżeli wzorzec rozpoczyna się od znaków innych niż litery.



# Indeksy oparte o haszowanie

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Indeksy oparte o haszowanie mogą być wykorzystywane tylko w przypadku porównywania wartości (operator =).

Utworzenie indeksu z haszowaniem:

```
CREATE INDEX name ON table USING hash (column);
```

GiST (ang. *Generalized Search Trees*) to infrastruktura, w ramach której mogą być implementowane różne strategie indeksowania dla różnych danych. Przykładowo, standardowo dostępne są indeksy dla danych dwuwymiarowych, wspierające zapytania z następującymi operatorami:

<<, >>

&<, &>

<<|, |>>

&<|, |&>

@>, <@

~=

&&

```
SELECT * FROM places ORDER BY  
location <-> point '(101,456)' LIMIT 10;
```

Indeksy odwrotne, które pozwalają na obsługę wartości zawierających więcej niż jeden klucz (nieatomicznych), takich jak tablice (*arrays*).

GIN, podobnie jak GiST, może korzystać z różnych strategii indeksowania i operatorów. Przykładowo, standardowo dostępne są operatory dla tablic:

<@

@>

=

&&

# Indeksy wielokolumnowe – przykład

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Weźmy tabelę:

```
CREATE TABLE test2 (  
    major int,  
    minor int,  
    name varchar  
);
```

Założmy, że często wykonywane są zapytania typu:

```
SELECT name FROM test2 WHERE major = constant  
AND minor = constant;
```

Możemy skorzystać z indeksu wielokolumnowego:

```
CREATE INDEX test2_mm_idx ON test2 (major, minor);
```

# Indeksy wielokolumnowe oparte o B-drzewa

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Indeksy wielokolumnowe są wspierane dla B-drzew, GiST oraz GIN i mogą zawierać do 32 kolumn.

Indeks oparty o B-drzewa, którego klucz wyszukiwania zawiera kilka kolumn może być stosowany dla dowolnego podzbioru tych kolumn, ale najbardziej wydajny będzie dla *kolumn wiodących*:

*Części indeksu do przejrzania zostaną wybrane na podstawie ograniczeń równości na kolumnach wiodących oraz ograniczeń nierówności na pierwszej kolumnie która nie posiada ograniczenia równości.*

# Indeksy wielokolumnowe oparte o GiST i GIN

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

- Wielokolumnowy indeks GiST również może być użyty z warunkami obejmującymi podzbiór jego klucza wyszukiwania.
- Warunek dotyczący pierwszej kolumny ma największy wpływ na wybór obszaru indeksu do przejrzania.
- Indeks GiST będzie mało wydajny, jeżeli pierwsza kolumna zawiera niewiele różnych wartości.
- Wielokolumnowy indeks GIN także obsługuje warunki dotyczące podzbioru klucza wyszukiwania, przy czym tu wybór podzbioru nie ma wpływu na wydajność.

# Łączenie indeksów

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Indeks wielokolumnowy może być wykorzystany, jeżeli ograniczenia nałożone na jego kolumny powiązane są operatorem AND.

PostgreSQL ma jednak możliwość łączenia w jednym zapytaniu kilku indeksów, a w szczególności wykorzystania jednego indeksu dla kilku różnych wartości.

Przy skanowaniu poszczególnych indeksów budowane są w pamięci *bitmapy* określające pasujące rekordy. Bitmapy te są następnie łączone zgodnie z zapytaniem, tzn. AND lub OR. Decyzja, czy korzystać z indeksów prostych czy złożonych, musi być oparta o analizę spodziewanego wykorzystania bazy.

# Indeksy uporządkowane

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Indeks może być w stanie dostarczać wyniki (wiersze spełniające warunek) w określonym porządku. Wtedy klauzula `ORDER BY` operacji nie wymaga dodatkowego kroku sortowania.

*Porządek można określić tylko dla indeksów opartych o B-drzewa.*

Planer przetwarza zapytanie z `ORDER BY` albo dopasowując indeks, albo przeglądając całą tabelę i sortując wyniki. Jeżeli zapytanie wymaga przejrzenia dużej części tabeli, przejście rekordów w kolejności fizycznej i posortowanie może wymagać mniej operacji I/O na dysku.



# Indeksy uporządkowane oparte o B-drzewa

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Domyślnie wartości klucza wyszukiwania sortowane są rosnąco, z nullami na końcu. Zachowanie to można zmienić:

```
CREATE INDEX test2_info_nulls_low
           ON test2 (info NULLS FIRST);

CREATE INDEX test3_desc_index
           ON test3 (id DESC NULLS LAST);
```

Indeks może być wykorzystany dla zapytań z klauzulą ORDER BY identyczną bądź całkowicie odwrotną.

# Indeksy unikalne

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Indeks może wymuszać unikalność wartości w kolumnie, bądź unikalność kombinacji wartości w kolumnach (w przypadku indeksu złożonego).

```
CREATE UNIQUE INDEX name ON table (column [, ...]);
```

Wartości NULL nie są uważane za identyczne.

PostgreSQL automatycznie tworzy unikalny indeks gdy definiowany jest klucz główny lub ograniczenie UNIQUE.

# Indeksy na wyrażeniach

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Klucz wyszukiwania nie musi składać się tylko z kolumn, może również obejmować wyrażenia, np. dla zapytania:

```
SELECT * FROM test1 WHERE lower(col1) = 'value';
```

przydatny może być indeks

```
CREATE INDEX test1_lower_col1_idx  
ON test1 (lower(col1));
```

Można też indeksować konkatelowane stringi:

```
CREATE INDEX people_names ON people  
((first_name || , , || last_name));
```

# Indeksy częściowe

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawansowane

Indeks może obejmować tylko część rekordów z tabeli.

```
CREATE TABLE access_log (  
    url varchar,  
    client_ip inet,  
    ...  
);
```

Gdybyśmy chcieli wyłączyć z indeksu lokalną podsieć, możemy go zdefiniować jako:

```
CREATE INDEX access_log_client_ip_ix  
    ON access_log (client_ip)  
WHERE NOT (client_ip > inet '192.168.100.0' AND  
    client_ip < inet '192.168.100.255');
```

# Porządki sortowania

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Indeks wspiera tylko jeden porządek sortowania (ang. *collation*) dla kolumny:

```
CREATE TABLE test1c (  
    id integer,  
    content varchar COLLATE "x"  
);
```

```
CREATE INDEX test1c_content_index  
    ON test1c (content);
```

Zapytania typu:

```
SELECT * FROM test1c WHERE content > constant;
```

działają, bo stała jest automatycznie rzutowana na odpowiedni porządek sortowania.

## Porządki sortowania, c.d.

Indeksowanie  
w bazach  
danych

Sebastian  
Ernst

Indeksy

Indeksy  
wielokolum-  
nowe

Indeksy  
uporząd-  
kowane

Zagadnienia  
zaawan-  
sowane

Indeks jednak nie zadziała dla stałej o innym porządku sortowania, np:

```
SELECT * FROM test1c
      WHERE content > constant COLLATE "y";
```

Możliwe jest jednak zbudowanie indeksu dla innego porządku sortowania:

```
CREATE INDEX test1c_content_y_index
      ON test1c (content COLLATE "y");
```