

Rozdział 18

Indeksy

Indeksy B-drzewo i bitmapowe, zwykłe i złożone, unikalne i nieunikalne, odwrócone, funkcyjne, skompresowane, bitmapowe połączeniowe.
Zarządzanie indeksami.



Indeksy

- dodatkowe struktury służące przyspieszaniu dostępu do danych,
- tworzone dla relacji, są jednak niezależne logicznie i fizycznie od danych relacji
- o użyciu indeksu przy realizacji operacji decyduje SZBD
- są automatycznie pielęgnowane przez bazę danych
- zalety:
 - przyspieszają odczyt danych (nie zawsze!),
 - wpływają na zwiększenie stopnia współbieżności wykonywanych w bazie danych operacji,
- wady:
 - mogą znacznie spowolnić operacje modyfikacji danych,
 - zajmują przestrzeń dyskową,



Struktura indeksu

- składa się z rekordów,
- rekord złożony jest z dwóch pól:
 - klucz – zawiera wartości występujące w atrybutach relacji, na których założono indeks, tzw. atrybutach indeksowych, lub wartości wyrażeń, zbudowanych z atrybutów relacji,
 - wskaźnik – określa blok zawierający rekordy, których wartości atrybutów indeksowych są równe wartościom klucza. W SZBD Oracle wskaźnik jest implementowany w postaci adresu rekordu (ang. rowid)



Adres rekordu

- określa dokładną fizyczną lokalizację rekordu w bazie danych,
- struktura: OOOOOOFFFFBBBBBBRRR:
 - OOOOOO – numer identyfikacyjny obiektu bazy danych (np. relacji), w której znajduje się rekord,
 - FFF – numer pliku bazy danych,
 - BBBBBB – numer bloku w pliku,
 - RRR – numer rekordu w bloku,
- odczyt adresu rekordu:

```
SELECT nazwisko, etat, rowid FROM pracownicy;
```
- adres rekordu jest niezmienny.



Użycie indeksu

1. użytkownik wykonuje zapytanie z warunkiem zawierającym poindeksowany atrybut,
2. SZBD szuka w indeksie klucza (zbioru kluczy), którego wartość (wartości) odpowiada wartości poindeksowanego atrybutu w warunku zapytania,
3. SZBD odczytuje adres rekordu (zbiór adresów rekordów) ze znajdującego w kroku 2. klucza (kluczy),
4. SZBD odczytuje rekord (zbiór rekordów), którego adresy (adresy) odczytał w kroku 3.



Jakie atrybuty indeksować?

- atrybuty często używane w klauzulach WHERE zapytań,
- atrybuty często używane w warunkach połączeniowych,
- atrybuty rzadko modyfikowane,
- atrybuty będące kluczami obcymi relacji,



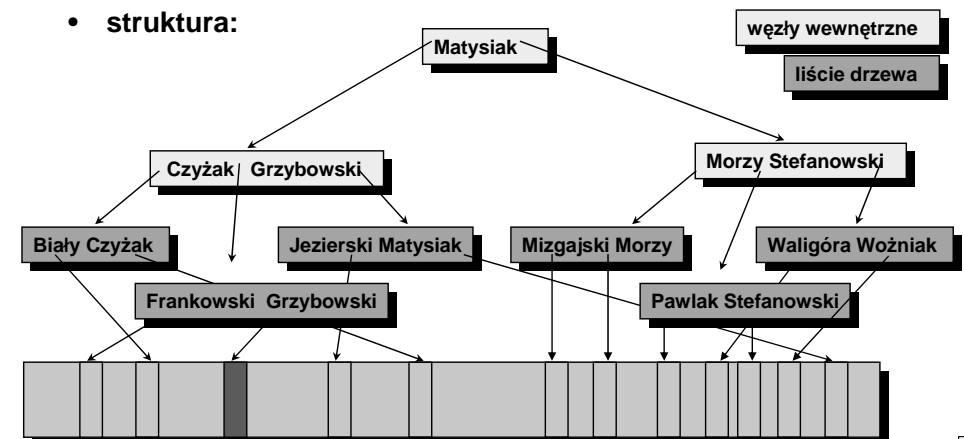
Podział indeksów

- ze względu na strukturę:
 - B-drzewa, bitmapowe,
- ze względu na liczbę atrybutów indeksowych w kluczu:
 - indeksy zwykłe i indeksy złożone,
- ze względu na unikalność wartości klucza:
 - indeksy unikalne i indeksy nieunikalne,
- ze względu na kolejność wartości klucza:
 - indeksy zwykłe i indeksy odwrócone,
- ze względu na sposób składowania:
 - indeksy nieskompresowane i indeksy skompresowane,
- ze względu na zastosowania:
 - indeksy funkcyjne i bitmapowe indeksy połączeniowe.



Indeks typu B-drzewo

- najczęściej stosowany w systemach OLTP (np. systemach obsługi bieżącej),
- definiowany tylko dla atrybutów o dużej selektywności,
- struktura:



Indeks typu B-drzewo (cd)

- **cechy:**
 - z kluczem indeksowym przechowywana lista adresów rekordów, w których wartości atrybutów indeksowych są równe wartości klucza,
 - wielkość indeksu słabo zależna od rozmiaru dziedziny atrybutu indeksowego,
 - efektywne wykonywanie operacji: koniunkcji, zapytań równościowych i przedziałowych, sortowania, testowania unikalności atrybutu, wyliczania wartości minimalnej i maksymalnej, grupowania, eliminacji powtórzeń,
 - wysoka współbieżność modyfikacji,
 - niski koszt pojedynczej modyfikacji, wysoki koszt modyfikacji grupy rekordów,
 - rozmiar indeksu może znacznie przewyższać rozmiar danych relacji,



Indeks typu B-drzewo (cd)

- **cechy (cd):**
 - nie przechowuje informacji o wartościach pustych,
 - dla relacji połączonych kluczem obcym eliminuje konieczność blokady relacji podrzędnej (tej, w której zdefiniowano klucz) w przypadku usuwania lub modyfikacji rekordów relacji nadrzędnej (tej, na którą wskazuje klucz),
- **składnia polecenia:**

```
CREATE INDEX nazwa ON relacja(atrybut);
```

- **przykład:**

```
CREATE INDEX nazwisko_idx ON pracownicy(nazwisko);  
CREATE INDEX placa_pod_idx ON pracownicy(placa_pod);
```



Indeks bitmapowy

- stosowany najczęściej w systemach OLAP (np. hurtowniach danych),
- definiowany tylko dla atrybutów o małej selektywności, atrybuty kandydujące:
 - liczba różnych wartości atrybutu powinna być mniejsza niż 1% liczby rekordów w relacji, lub
 - wartości atrybutu powtarzają się ponad 100 razy w relacji.
- **cechy:**
 - z kluczem przechowywana bitmapa, której pozycje odpowiadają adresom rekordów o wartościach atrybutów indeksowych równych wartości klucza,
 - konwersja pozycji bitmapy na adres rekordu realizowana przez funkcję mapującą,
 - stosowane w zapytaniach z warunkami z operatorem „=”
 - wielkość indeksu silnie zależna od rozmiaru dziedziny atrybutu indeksowego,



Indeks bitmapowy (cd)

- **cechy (cd.):**
 - efektywne wykonywanie operacji koniunkcji, alternatywy i negacji,
 - wykorzystywany w zapytaniach z poszukiwaniem wartości pustych,
 - niska współbieżność modyfikacji – konieczność blokady całej mapy bitowej,
 - wysoki koszt pojedynczej modyfikacji, niski koszt modyfikacji grupy rekordów,
 - rozmiar indeksu jest najczęściej ułamkiem rozmiaru danych relacji,
- w poleceniu tworzenia indeksu dodatkowa klauzula BITMAP,
- **przykład:**

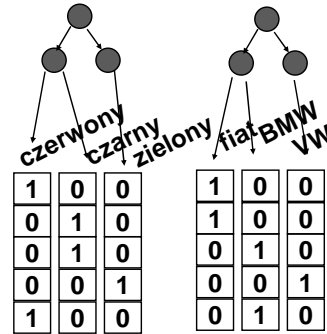
```
CREATE BITMAP INDEX prac_plec_bmp_idx ON pracownicy (plec);
```



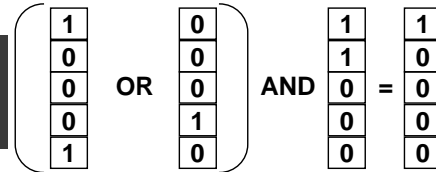
Indeks bitmapowy (cd)

| Nr rej | Kolor | Marka |
|----------|----------|-------|
| PWG01425 | czerwony | fiat |
| WAW3456 | czarny | fiat |
| POZ3756 | czarny | BMW |
| KTW3756 | zielony | VW |
| PNR8956 | czerwony | BMW |

```
CREATE BITMAP INDEX marka_bmp_idx
ON samochody (marka);
CREATE BITMAP INDEX kolor_bmp_idx
ON samochody (kolor);
```



```
SELECT COUNT(*) FROM samochody
WHERE kolor IN ('czerwony', 'zielony')
AND marka = 'fiat';
```



Indeks złożony

- klucz indeksu zawiera więcej niż jeden atrybut relacji,
- maksymalnie 32 atrybuty w kluczu indeksu (30 dla indeksu bitmapowego),
- dla indeksu złożonego na atrybutach ABC kombinacje atrybutów: A, AB i ABC to tzw. części wiodące klucza, w przeciwieństwie do kombinacji B, BC oraz C,
- przykład:

```
CREATE INDEX nazw_etat_idx ON pracownicy(nazwisko, etat);
CREATE INDEX id_prac_etat_placa_idx
ON pracownicy(id_prac, etat, placa_pod);
```

Indeks złożony (cd)

- kiedy zakładać indeks złożony:
 - na atrybutach często występujących razem w klauzuli WHERE zapytań
 - na atrybutach często odczytywanych wspólnie przez wiele zapytań,
- jak wybrać kolejność atrybutów w kluczu:
 - atrybuty wykorzystywane w klauzuli WHERE powinny stanowić część wiodącą klucza,
 - atrybuty wykorzystywane częściej w klauzuli WHERE powinny stanowić część wiodącą klucza,
 - jeśli częstotliwość atrybutów jest ta sama, pierwszym atrybutem powinien być ten, wg którego wartości danych są fizycznie posortowane.

Indeks unikalny i nieunikalny

- indeks unikalny – gwarantuje, że w relacji nie będzie dwóch rekordów z tą samą wartością atrybutu indeksowego (atrybutów indeksowych w przypadku indeksu złożonego), w przeciwieństwie do indeksu nieunikalnego,
- w SZBD Oracle indeksy unikalne są tworzone automatycznie przy definiowaniu ograniczeń integralnościowych typu klucz podstawowy i klucz unikalny,
- w poleceniu tworzenia indeksu dodatkowa klauzula UNIQUE,
- przykład:

```
CREATE UNIQUE INDEX id_prac_idx ON pracownicy(id_prac);
CREATE UNIQUE INDEX id_zesp_idx ON zespoly(id_zesp);
```

- Uwaga! Nie można utworzyć unikalnego indeksu bitmapowego!

Indeks funkcyjny

- definiowany dla atrybutów, które w zapytaniach często używane są jako parametry funkcji (np. upper(nazwisko)) bądź elementy wyrażeń (np. placa_pod * 1.2),
- może być zaimplementowany jako indeks B-drzewo lub indeks bitmapowy,
- SZBD nie użyje indeksu niefunkcyjnego, założonego na atrybucie A, gdy w zapytaniu A jest parametrem funkcji lub elementem wyrażenia,
- indeksowane wyrażenie nie może zawierać funkcji agregujących,
- przykład:

```
SELECT * FROM pracownicy  
WHERE UPPER(nazwisko) = 'NOWAK'  
AND placa_dod*2 = placa_pod;
```

```
CREATE INDEX nazw_idx_fun ON pracownicy(UPPER(nazwisko));  
CREATE INDEX placa_dod_idx_fun ON pracownicy(placa_dod*2);
```

Indeks odwrócony

- wartości w kluczu indeksowym składowane są w postaci odwróconej,

| wartość oryginalna | wartość składowana |
|--------------------|--------------------|
| 802121 | 121208 |
| 802122 | 221208 |
| 802123 | 321208 |

- stosowane do indeksowania sekwencji, powodują rozproszenie wartości w indeksie,
- w poleceniu tworzenia indeksu dodatkowa klauzula REVERSE,
- przykład:

```
CREATE INDEX lp_rev_idx ON pracownicy(lp) REVERSE;
```

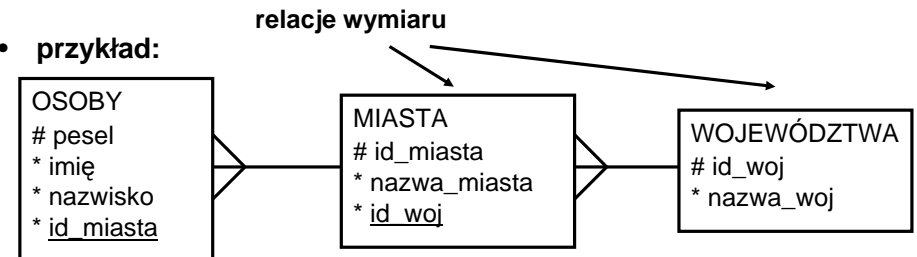
Bitmapowy indeks połączeniowy

- indeks definiowany dla operacji równościowego połączenia dwóch lub więcej relacji,
- dla każdej wartości atrybutu indeksowego relacji składowane są adresy rekordów drugiej relacji, które mają tę samą wartość atrybutu połączeniowego,
- wykorzystywany przy zapytaniach łączących relacje,
- składnia:

```
CREATE BITMAP INDEX nazwa ON relacja (lista_atrybutów)  
FROM relacja_1, relacja_2, ..., relacja_n  
WHERE warunek_połączeniowy_1 AND warunek_połączeniowy_2 ...  
AND warunek_połączeniowy_n-1;
```

Bitmapowy indeks połączeniowy (cd)

- przykład:



```
SELECT COUNT(*)  
FROM osoby NATURAL JOIN miasta NATURAL JOIN wojewodztwa  
WHERE nazwa_woj = 'Wielkopolskie';
```

```
CREATE BITMAP INDEX os_mi_woj_bmp_idx  
ON osoby(nazwa_woj)  
FROM osoby o, miasta m, wojewodztwa w  
WHERE o.id_miesta = m.id_miesta AND m.id_woj = w.id_woj;
```

Bitmapowy indeks połączeniowy (cd)

- ograniczenia:
 - relacja nie może pojawiać się wielokrotnie w złączeniu,
 - przy tworzeniu indeksu w definicji połączenia nie można używać składni ANSI,
 - poindeksowane atrybuty muszą być atrybutami relacji wymiarów,
 - atrybuty relacji wymiarów, umieszczone w warunku połączeniowym, muszą mieć zdefiniowane ograniczenia typu klucz podstawowy lub klucz unikalny



Kompresja indeksu

- redukuje zajętości przestrzeni dyskowej przez powtarzające się wartości klucza indeksu,
- ograniczenia:
 - tylko dla indeksów typu B-drzewo,
 - dla indeksów nieunikalnych wszystkie atrybuty klucza mogą zostać skompresowane,
 - dla indeksów unikalnych przynajmniej jeden z atrybutów klucza musi pozostać nieskompresowany,
- zalety:
 - duża oszczędność przestrzeni dyskowej – większa liczba kluczy indeksu składowana w bloku,
 - zwiększenie wydajności operacji we/wy z użyciem indeksu,
- wady:
 - większe zużycie CPU w celu dekompresji kluczy przy przeglądaniu indeksu,



Kompresja indeksu (cd)

- domyślnie indeksy są tworzone jako indeksy bez kompresji,
- dodatkowa klauzula **COMPRESS n**, gdzie n to liczba atrybutów w kluczu, które mają zostać skompresowane,
- przykład:

```
CREATE INDEX prac_idx ON pracownicy(id_zesp, etat)
COMPRESS 1;
```



Zarządzanie indeksami

- Usunięcie indeksu:

```
DROP INDEX nazwa_indeksu;
```

- Przebudowa indeksu:

```
ALTER INDEX nazwa_indeksu REBUILD;
```

- Wyliczenie statystyk dla indeksu:

```
CREATE INDEX ... COMPUTE STATISTICS;
ALTER INDEX nazwa_indeksu COMPUTE STATISTICS;
```

- Zmiana nazwy indeksu:

```
ALTER INDEX nazwa_indeksu RENAME TO nowa_nazwa;
```



Słownik danych

| | |
|-----------------------|--|
| USER_INDEXES | informacje o wszystkich indeksach, będących własnością użytkownika (synonim IND) |
| USER_IND_COLUMNS | informacje o poindeksowanych atrybutach |
| USER_IND_EXPRESSIONS | informacje o wyrażeniach, na których zbudowano indeksy funkcyjne |
| USER_JOIN_IND_COLUMNS | informacje o atrybutach w warunkach połączeniowych dla indeksów połączeniowych |

```
SELECT i.index_name, i.index_type, i.uniqueness, c.column_name
FROM user_indexes i, user_ind_columns c
WHERE i.table_name = 'PRACOWNICY'
AND i.index_name = c.index_name
ORDER BY i.index_name, c.column_position;
```

Relacja zorganizowana jak indeks (IOT)

- dane relacji przechowywane w strukturze B-drzewa,
- uporządkowanie danych w drzewie wg wartości atrybutów klucza podstawowego relacji, rekord jest identyfikowany przez wartość klucza podstawowego a nie przez adres rekordu (ROWID),
- w liściu B-drzewa znajduje się cały rekord relacji, dla relacji o dużym rozmiarze rekordu można zdefiniować dodatkowe miejsce składowania, tzw. obszar przepełnienia, poza liśćmi B-drzewa dla atrybutów niekluczowych,
- zalety:
 - szybszy dostęp do danych w zapytaniach z warunkiem zbudowanym z atrybutów klucza podstawowego,
 - brak konieczności zakładania indeksów na atrybutach klucza podstawowego – oszczędność przestrzeni dyskowej,
 - na atrybutach niekluczowych można zdefiniować dodatkowe indeksy (dzięki istnieniu logicznego ROWID),

Relacja zorganizowana jak indeks (IOT) (cd)

- ograniczenia:
 - IOT musi mieć zdefiniowany klucz podstawowy,
 - polecenie modyfikacji wartości klucza podstawowego relacji może wymagać przebudowy całej struktury,
- składnia polecenia:

```
CREATE TABLE nazwa (
    definicja atrybutów
    definicja ograniczeń integralnościowych)
ORGANIZATION INDEX
[PCTTHRESHOLD procent]
[OVERFLOW TABLESPACE przestrzeń tabel [INCLUDING lista
atrybutów]];
```

- informacje w słowniku: USER_TABLES