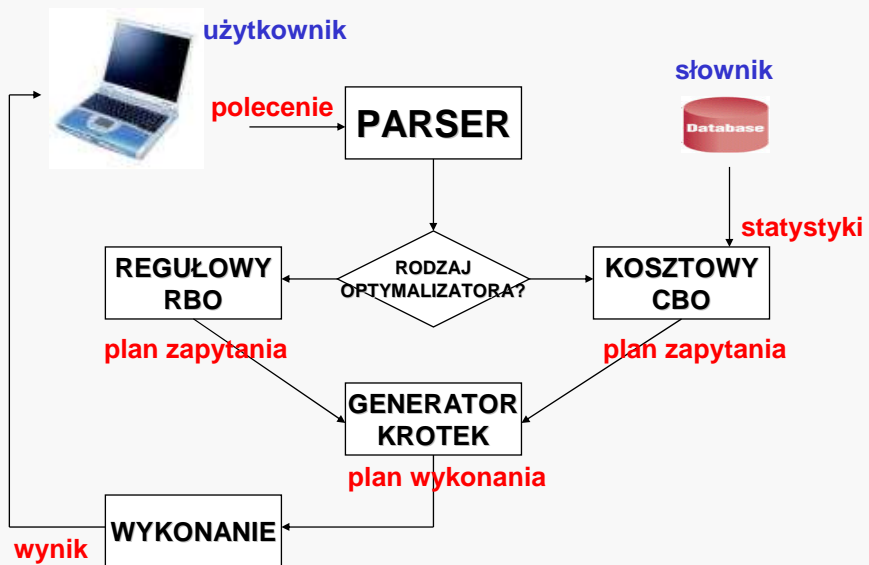


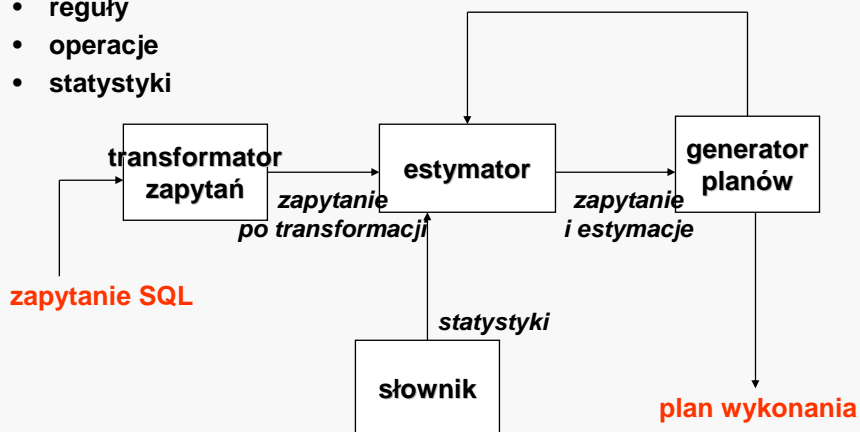
Optymalizacja poleceń SQL

Przetwarzanie polecenia SQL



Optymalizacja polecenia SQL

- struktury danych
- metody dostępu
- reguły
- operacje
- statystyki



(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

3

Optymalizacja

Optymalizacja to proces doboru odpowiednich struktur danych, metod dostępu i operacji (planu wykonania), w celu zminimalizowania kosztu realizacji polecenia.

Optymalizacja jest wykonywana przez wyspecjalizowany moduł systemu – optymalizator zapytań.

- **regułowa**
 - oparta na rankingu metod dostępu do struktur danych
 - preferowana dla aplikacji spadkowych
- **kosztowa**
 - oparta na szacowaniu kosztu (czas zajętości procesora, liczby operacji we/wy, zajętość pamięci operacyjnej itp.), wykonania wszystkich potencjalnych planów wykonania
 - zalecana dla wszystkich nowopowstających aplikacji
 - zakłada duże obciążenie systemu: dużą współbieżność operacji, niski współczynnik trafień w bufor danych

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

4

Statystyki

Informacje charakteryzujące struktury danych
dane generowane dla tabeli:

- liczba wierszy,
- liczba bloków danych zawierających dane,
- liczba nigdy nie użytych, zaalokowanych bloków danych,
- średnia wielkość wolnego miejsca w zajętych blokach danych,
- liczba łańcuchowanych wierszy,
- średnia wielkość wiersza,
- dla wszystkich kolumn liczbę unikalnych wartości oraz wartość minimalną i maksymalną

dane generowane dla indeksu:

- wysokość drzewa,
- liczbę bloków-liści drzewa,
- liczbę unikalnych wartości indeksu,
- średnią liczbę bloków-liści przypadającą na jedną wartość klucza indeksu,
- średnią liczbę bloków danych (w tabeli) przypadającą jedną wartość klucza indeksu,
- współczynnik zgrupowania, który określa na ile wiersze w tabeli są uporządkowane wg klucza indeksy.

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

5

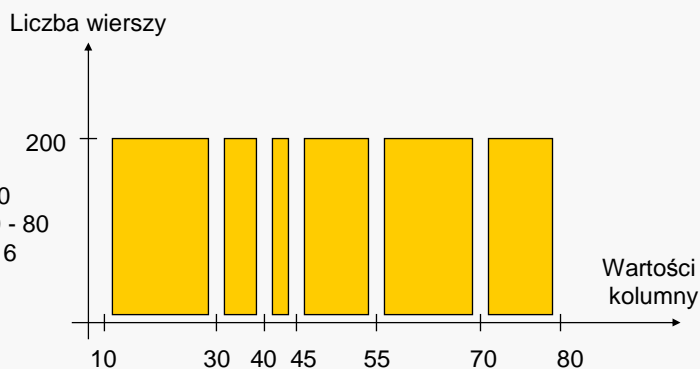
Histogramy

Szczegółowe statystyki opisujące rozkład wartości poszczególnych kolumn, przydatne w szczególności dla optymalizacji wykorzystania indeksów

Przykład:

Dane:

- liczba wierszy 1200
- zakres wartości 10 - 80
- liczba przedziałów 6

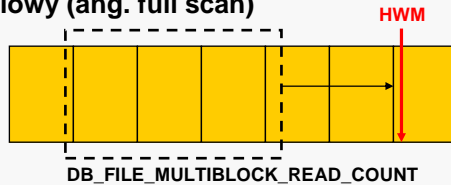


(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

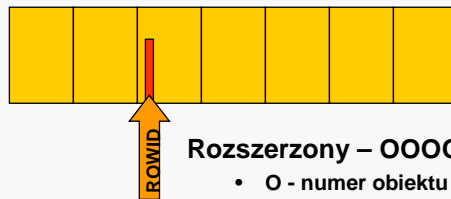
6

Metody dostępu do danych

Przeгляд liniowy (ang. full scan)



Dostęp za pomocą adresu rekordu (ang. ROWID)



Rozszerzony – OOOOOO.FFF.BBBBBB.RRR

- O - numer obiektu w bazie danych
- F - względny numer pliku w przestrzeni tabel
- B - numer bloku w pliku
- R - numer rekordu w bloku

Podstawowy – FFFF.BBBBBBBB.RRRR

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

7

Rodzaje sortowania

- **ORDER BY:** sortowanie wyników zapytania

```
SELECT * FROM zespolo
ORDER BY adres DESC;
```

- **AGGREGATE:** wyliczanie wartości funkcji grupowej

```
SELECT MAX(zatrudniony)
FROM pracownicy;
```

- **GROUP BY:** podział relacji na grupy

```
SELECT etat, AVG(placa_pod)
FROM pracownicy GROUP BY etat;
```

- **UNIQUE:** eliminacja duplikatów

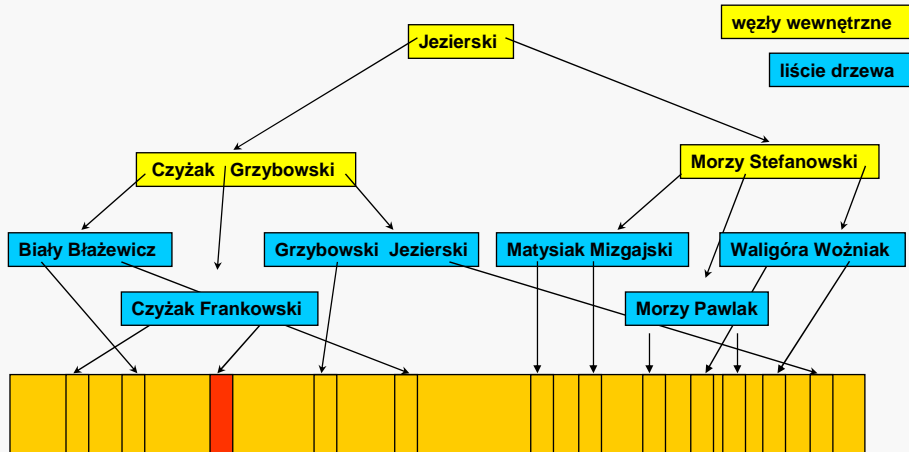
```
SELECT DISTINCT etat
FROM pracownicy;
```

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

8

Indeks B*-drzewo

```
SELECT ETAT, PLACA_POD  
FROM PRACOWNICY  
WHERE NAZWISKO = 'Frankowski';
```



(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

9

Rodzaje B*-drzew

- Unikalność wartości kluczy: nieunikalne, unikalne
- Liczba atrybutów w kluczu: zwykłe, skonkatelowane
- Kompresja: nie skompresowane, skompresowane
- Porządek kluczy: leksykograficzny, odwrócony

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

10

Przesłanki do utworzenia indeksu B*-drzewo

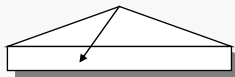
- na atrybutach często wykorzystywanych w warunkach selekcji,
- na atrybutach połączeniowych,
- tylko na atrybutach o dużej selektywności,
- na atrybutach rzadko modyfikowanych,
- na atrybutach będących kluczami obcymi (uniknięcie niepotrzebnego blokowania tabeli podrzędnej w przypadku operacji modyfikacji rekordów nadrzędnych)
- w systemach przetwarzania transakcyjnego - OLTP

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

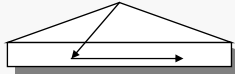
11

Metody dostępu do indeksu B*- drzewo

unikalne przeglądnięcie (ang. **unique scan**)



przeglądnięcie zakresu (ang. **range scan**)



pełne przeglądnięcie (ang. **full scan**)



- odczyt blok po bloku - nawigacja po liściach
- stosowany również do sortowania

szybkie pełne przeglądnięcie (ang. **fast full scan**)



- odczyt wieloblokowy
- stosowany zamiast full table scan

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

12

Połączenia - nested loop

Odczyt
jednokrotny



Tabela
zewnątrza

| | |
|---|---|
| A | 3 |
| B | 2 |
| C | 1 |
| D | 3 |

Tabela
wewnętrzna

| | |
|---|---|
| 2 | a |
| 1 | b |
| 2 | c |
| 3 | d |
| 1 | e |

Odczyt
wielokrotny



| | | | |
|---|---|---|---|
| A | 3 | 3 | d |
| B | 2 | 2 | a |
| B | 2 | 2 | c |
| C | 1 | 1 | b |
| C | 1 | 1 | e |
| D | 3 | 3 | d |

Wynik
połączenia

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

13

Połączenia - sort merge



| | |
|---|---|
| A | 3 |
| B | 2 |
| C | 1 |
| D | 3 |

| | |
|---|---|
| 2 | a |
| 1 | b |
| 2 | c |
| 3 | d |
| 1 | e |



sortowanie



| | |
|---|---|
| C | 1 |
| B | 2 |
| A | 3 |
| D | 3 |

| | |
|---|---|
| 1 | b |
| 1 | e |
| 2 | a |
| 2 | c |
| 3 | d |



złączenie

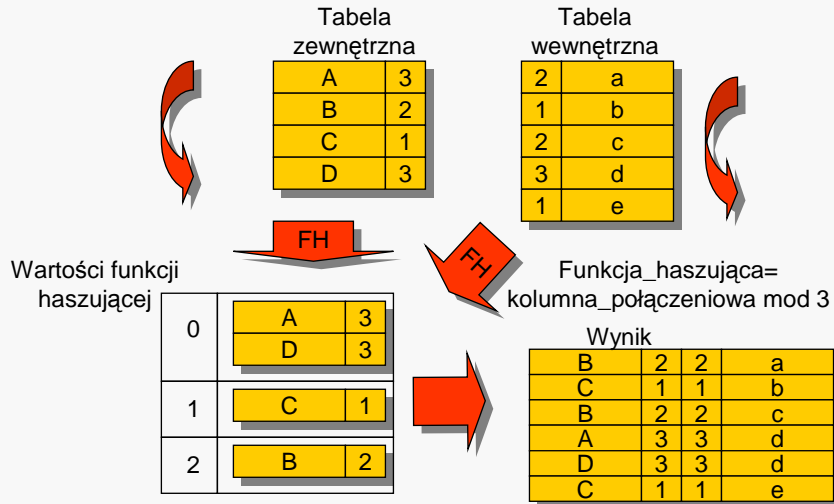


| | | | |
|---|---|---|---|
| C | 1 | 1 | b |
| C | 1 | 1 | e |
| B | 2 | 2 | a |
| B | 2 | 2 | c |
| A | 3 | 3 | d |
| D | 3 | 3 | d |

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

14

Połączenia - hash join



(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

15

Wybór optymalizatora

```
ALTER SESSION SET OPTIMIZER_MODE='RULE';
```

```
ALTER SESSION SET OPTIMIZER_MODE='CHOOSE';
```

```
DBMS_STATS.GATHER_SCHEMA_STATS
DBMS_STATS.GATHER_TABLE_STATS
DBMS_STATS.GATHER_INDEX_STATS
```

```
exec dbms_stats.gather_schema_stats(ownname=>'DEMO92')
exec dbms_stats.gather_table_stats(ownname=>'SCOTT', -
tabname=>'KLIENCI')
```

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

16

Analiza planu wykonania zapytania

```
explain plan set statement_id='identyfikator'  
for wyjaśniane_polecenie;
```

```
explain plan set statement_id='1'  
for  
select sk.nazwa, sp.ilosc  
from sklepy sk, sprzedaz@lab92 sp  
where sk.sklep_id=sp.sklep_id;
```

```
set autotrace {on | traceonly} explain
```

```
set autotrace off
```

Analiza planu wykonania zapytania (2)

```
select sk.nazwa, sp.ilosc  
from sklepy sk, sprzedaz@lab92 sp  
where sk.sklep_id=sp.sklep_id  
and sk.miasto='Poznań';
```

| OPERATION | OPTIONS | OBJECT_NAME | OBJECT_NODE | OTHER |
|------------------|----------|-------------|-------------|--|
| ----- | | | | |
| SELECT STATEMENT | | | | |
| NESTED LOOPS | | | | |
| REMOTE | | | | LAB92.WORLD SELECT "ILOSC", "SKLEP_ID" FROM "SPRZEDAZ" "SP" |
| TABLE ACCESS | BY INDEX | SKLEPY | | |
| | ROWID | | | |
| INDEX | UNIQUE | SKLEPY_PK | | |
| | SCAN | | | |

Analiza planu wykonania zapytania (3)

```
select kl.nazwisko, kl.miasto, sp.ilosc, pr.nazwa
from klienci kl, sprzedaz@lab92 sp, produkty@lab92 pr
where kl.klient_id=sp.klient_id
and sp.produkt_id=pr.produkt_id and pr.produkt_id=30;
```

Execution Plan

```
-----
0 SELECT STATEMENT Optimizer=RULE
1  NESTED LOOPS
2    MERGE JOIN
3      SORT (JOIN)
4        REMOTE*                                LAB92.WORLD
5          SORT (JOIN)
6            REMOTE*                                LAB92.WORLD
7          TABLE ACCESS (BY INDEX ROWID) OF 'KLIENCI'
8            INDEX (UNIQUE SCAN) OF 'KLIENCI_PK' (UNIQUE)

4 SERIAL_FROM_REMOTE                          SELECT "PRODUKT_ID", "NAZWA" FROM
                                                "PRODUKTY" "PR" WHERE "PRODUKT_ID"=30
6 SERIAL_FROM_REMOTE                          SELECT "ILOSC", "PRODUKT_ID", "KLIENT_ID"
                                                FROM "SPRZEDAZ" "SP"
```

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

19

Zapytania rozproszone – agregaty

- wykorzystanie perspektywy

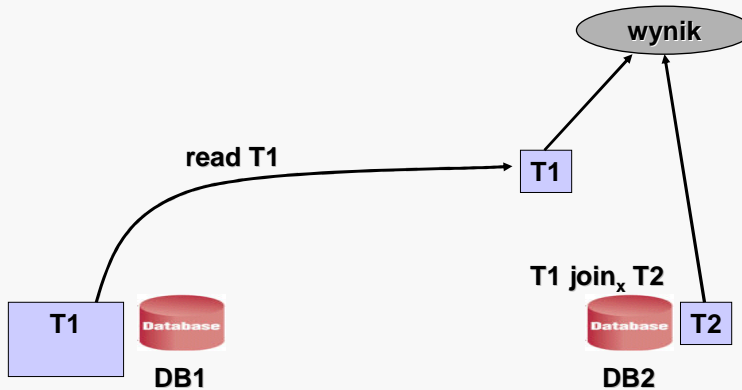


(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

20

Zapytania rozproszone – połączenia (1)

- wykonanie operacji $T_1 \text{ join}_x T_2$ w bazie DB_2
 - jeżeli T_1 jest duża wówczas wystąpi duży koszt transmisji sieciowej

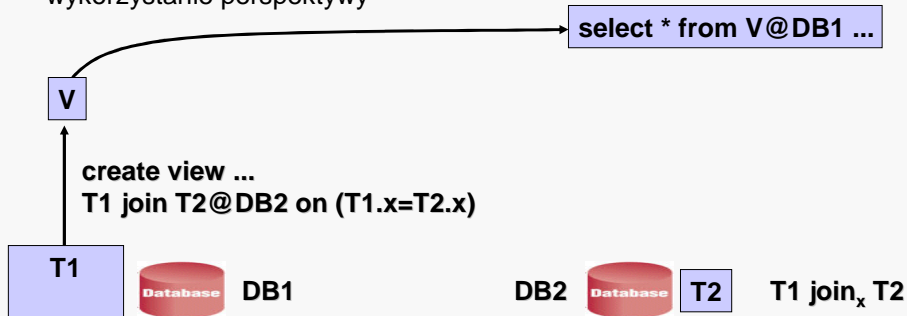


(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

21

Zapytania rozproszone – połączenia (3)

- wykorzystanie perspektywy



- zastąpienie sort-merge przez nested loops
 - w przypadku, gdy niewiele rekordów z T_1 łączy się z T_2

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

22

Wskazówki

ORDERED

zachowanie kolejności tabel, takiej jak w klauzuli FROM
tabelą zewnętrzną jest pierwsza z lewej

```
select /*+ ORDERED */ sk.nazwa, sp.ilosc
from sklepy sk, sprzedaz@lab92 sp
where sk.sklep_id=sp.sklep_id;
```

FULL

wskazanie tabeli odczytywanej za pomocą full table scan

```
select /*+ FULL(sp) */ sk.nazwa, sp.ilosc
from sprzedaz@lab92 sp, sklepy sk
where sk.sklep_id=sp.sklep_id;
```

Wskazówki (2)

DRIVING_SITE

wskazanie bazy danych, w której ma zostać wykonane zapytanie

```
select /*+ DRIVING_SITE(sp) */ kl.nazwisko, sp.data,
sp.ilosc, sp.produkt_id
from klienci kl, sprzedaz@lab92 sp
where kl.klient_id=sp.klient_id
and kl.miasto='Poznań';
```

Wskazówki (3)

NO_MERGE

zapytanie w perspektywie wbudowanej nie jest scalane z zapytaniem głównym

```
select /*+ NO_MERGE(v) */ kl.nazwisko, kl.miasto, v.ilosc,  
v.nazwa  
from klienci kl,  
  (select sp.klient_id, sp.ilosc, pr.nazwa, pr.produkt_id  
   from sprzedaz@lab92 sp, produkty@lab92 pr  
   where sp.produkt_id=pr.produkt_id) v  
where kl.klient_id=v.klient_id  
and v.produkt_id=30;
```

USE_HASH

USE_MERGE

USE_NL

wybór algorytmu łączenia tabel

(c) J.Jeziński, M.Morzy, R. Wrembel: Politechnika Poznańska, Instytut Informatyki

25

Środowisko testowe

