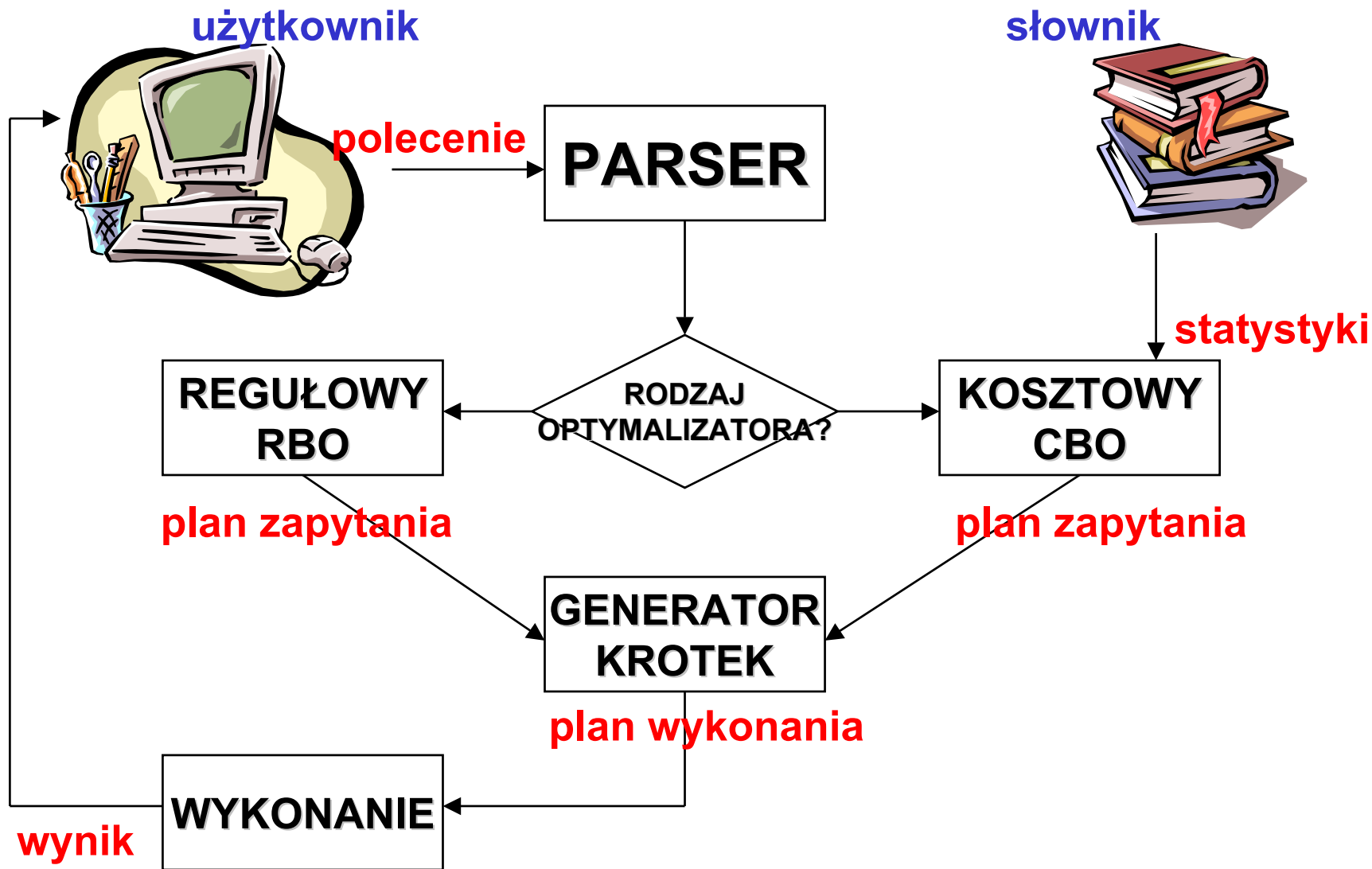


# Rozdział 18

## Optymalizacja poleceń SQL

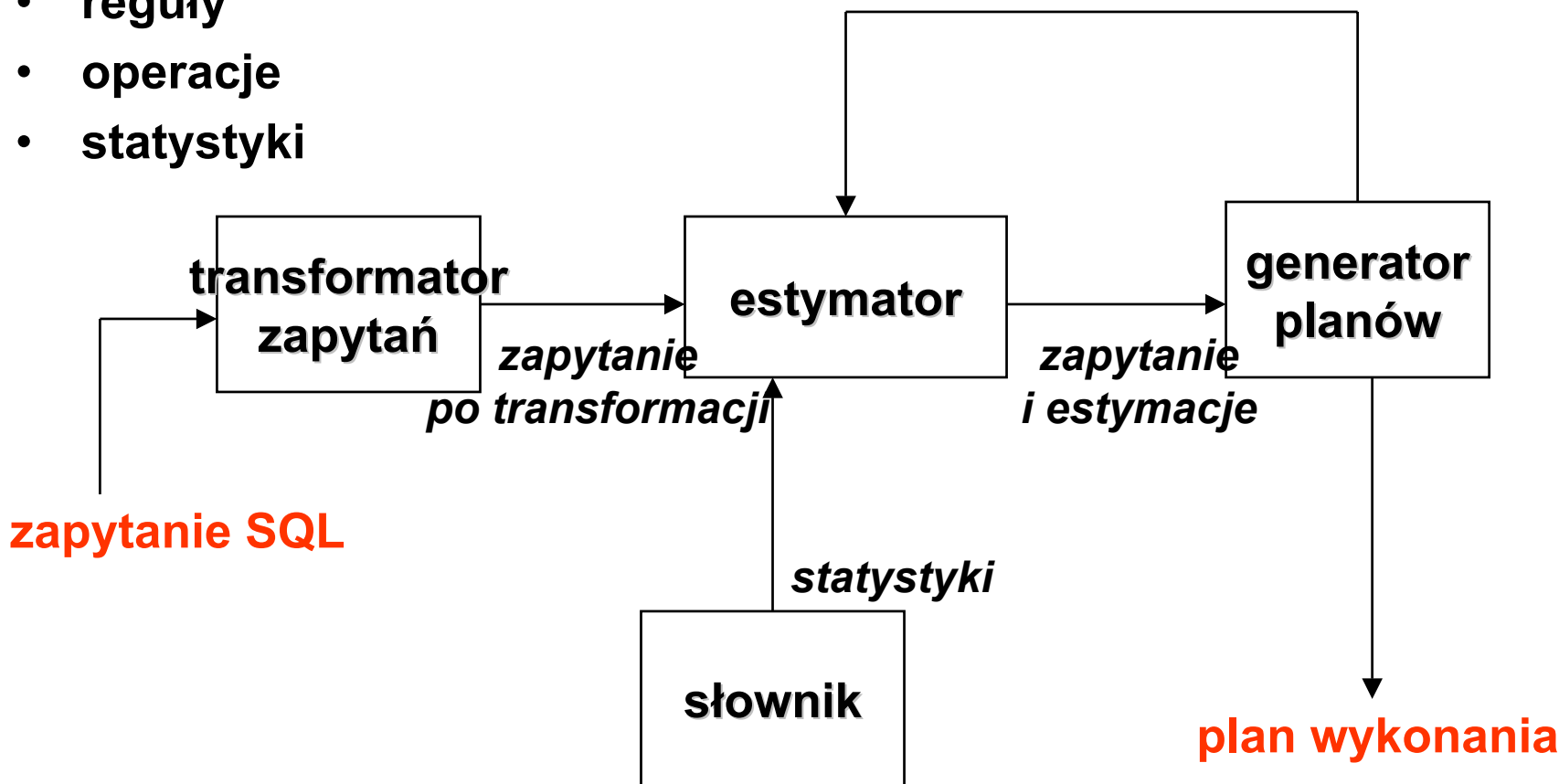
**Schemat optymalizacji, polecenie EXPLAIN  
PLAN, dyrektywa AUTOTRACE, optymalizacja  
kosztowa i regułowa, statystyki i histogramy,  
metody dostępu i sortowania, indeksy typu B\*  
drzewo, indeksy bitmapowe i funkcyjne, metody  
połączeń, wskazówki**

# Przetwarzanie polecenia SQL



# Optimalizacja polecenia SQL

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



# Polecenie EXPLAIN PLAN

## Utworzenie tabeli PLAN\_TABLE

```
SQL> @$ORACLE_HOME\RDBMS\ADMIN\UTLXPLAN.SQL
```

## Wygenerowanie planu wykonania zapytania

```
EXPLAIN PLAN  
[ SET STATEMENT_ID = 'identyfikator' ]  
FOR SELECT ... FROM ... WHERE ... ;
```

## Obejrzenie wyników

```
SQL> @$ORACLE_HOME\RDBMS\ADMIN\UTLXPLS.SQL
```

# Dyrektywa AUTOTRACE

```
SQL> @$ORACLE_HOME\SQLPLUS\ADMIN\PLUSTRCE.SQL  
SQL> GRANT PLUSTRACE TO SCOTT
```

```
SQL> SET AUTOTRACE [ ON | OFF ] [ TRACEONLY ]  
      [ EXPLAIN ] [ STATISTICS ]
```

```
SQL> SET AUTOTRACE TRACEONLY EXPLAIN
```

```
SQL> SELECT nazwisko FROM pracownicy WHERE id_prac=100;
```

Plan wykonania

```
-----  
 0      SELECT STATEMENT Optimizer=CHOOSE (Cost=1 Card=1 Bytes=22)  
 1    0    TABLE ACCESS (BY ROWID) OF 'PRACOWNICY' (Cost=1 Card=1 Bytes=22)  
 2    1      INDEX (UNIQUE SCAN) OF 'PK_PRAC' (UNIQUE)
```

```
SQL> SET AUTOTRACE OFF
```

# 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

# Zmiana celu optymalizacji

W pliku inicjalizacyjnym bazy danych

- parametr `OPTIMIZER_MODE = [ CHOOSE | RULE ]`

Dla bieżącej sesji

- parametr `OPTIMIZER_GOAL`
  - **CHOOSE**: kosztowa (jeśli są statystyki) lub regułowa
  - **RULE**: optymalizacja regułowa
  - **ALL\_ROWS**: kosztowa maksymalizująca przepustowość
  - **FIRST\_ROWS**: kosztowa minimalizująca czas odpowiedzi
  - **FIRST\_ROWS\_N**: kosztowa minimalizująca łączny czas odczytania pierwszych N (1,10,100,1000) krotek

```
ALTER SESSION SET OPTIMIZER_GOAL = FIRST_ROWS;
```

# 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.



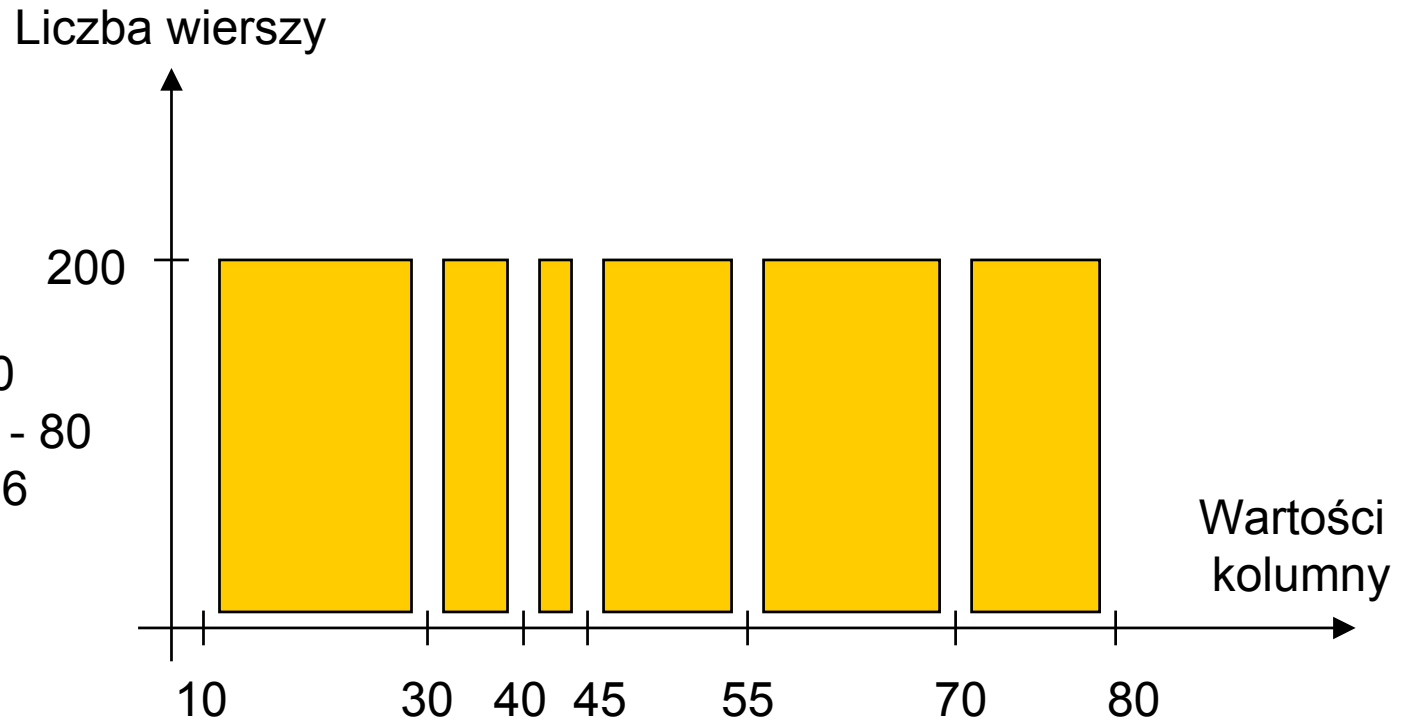
# Histogramy

**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



# Zbieranie statystyk (do Oracle 8.0)

```
ANALYZE TABLE nazwa_tabeli  
  COMPUTE STATISTICS  
  FOR TABLE FOR ALL INDEXES;
```

```
ANALYZE TABLE nazwa_tabeli  
  ESTIMATE STATISTICS  
  SAMPLE liczba ROWS | PERCENT;
```

```
ANALYZE TABLE nazwa_tabeli  
  DELETE STATISTICS;
```

```
ANALYZE TABLE zespoly COMPUTE STATISTICS;  
ANALYZE TABLE zespoly  
  ESTIMATE STATISTICS SAMPLE 20 PERCENT;
```

```
SELECT NUM_ROWS, BLOCKS, EMPTY_BLOCKS,  
  AVG_ROW_LEN, LAST_ANALYZED FROM USER_TABLES  
WHERE TABLE_NAME = 'ZESPOLY';
```

# Zbieranie histogramów (do Oracle 8.0)

```
ANALYZE TABLE nazwa_tabeli  
  COMPUTE STATISTICS  
  [ FOR ALL INDEXES ]  
  [ FOR ALL [ INDEXED ] COLUMNS [ SIZE liczba ] ] |  
  [ FOR COLUMNS [ SIZE liczba ] kolumna1 [SIZE liczba], ... ] } ;
```

```
ANALYZE TABLE pracownicy  
ESTIMATE STATISTICS SAMPLE 10 PERCENT  
FOR ALL INDEXES  
FOR COLUMNS etat SIZE 5, placa_pod SIZE 3;
```

```
SELECT COLUMN_NAME, NUM_DISTINCT, LOW_VALUE,  
       HIGH_VALUE, NUM_NULLS, AVG_COL_LEN  
FROM USER_TAB_COL_STATISTICS  
WHERE TABLE_NAME='PRACOWNICY';
```

# Zbieranie statystyk za pomocą pakietu DBMS\_STATS (Oracle 8i)

GATHER\_INDEX\_STATS - indeksu

GATHER\_TABLE\_STATS - tabela, indeks, kolumny

GATHER\_SCHEMA\_STATS - wszystkie obiekty w schemacie

GATHER\_DATABASE\_STATS - wszystkie obiekty w bazie danych

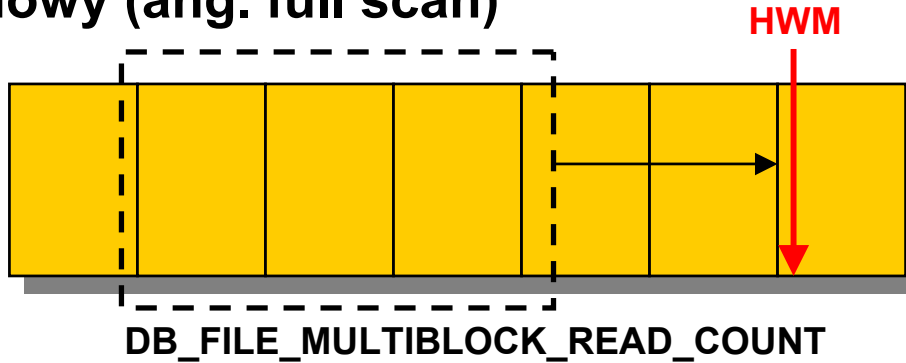
```
EXEC DBMS_STATS.GATHER_TABLE_STATS('MIKOLAJ','ZESPOLY');
```

```
EXEC DBMS_STATS.GATHER_TABLE_STATS(  
    ownname=>'MIKOLAJ',  
    tabname=>'PRACOWNICY',  
    estimate_percent=>25,  
    method_opt=>'FOR ALL COLUMNS');
```

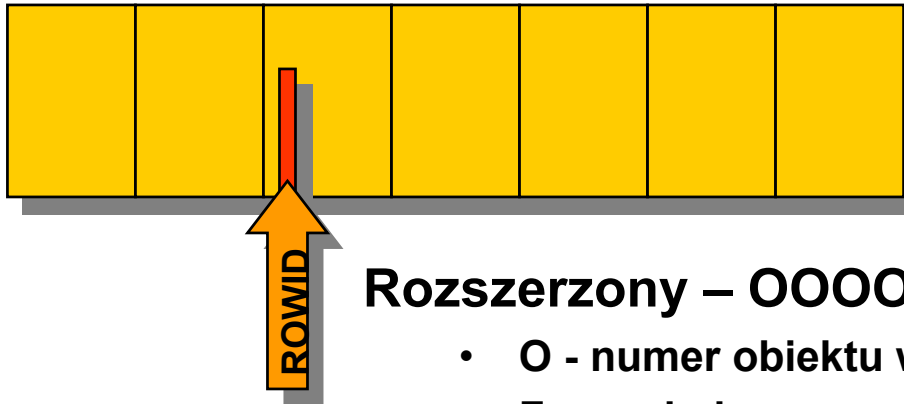
```
EXEC DBMS_STATS.DELETE_TABLE_STATS('MIKOLAJ','ZESPOLY');
```

# Metody dostępu do danych

## Przegląd 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**

# Rodzaje sortowania

- **ORDER BY:** sortowanie wyników zapytania

- **AGGREGATE:** wyliczanie wartości funkcji grupowej

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

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

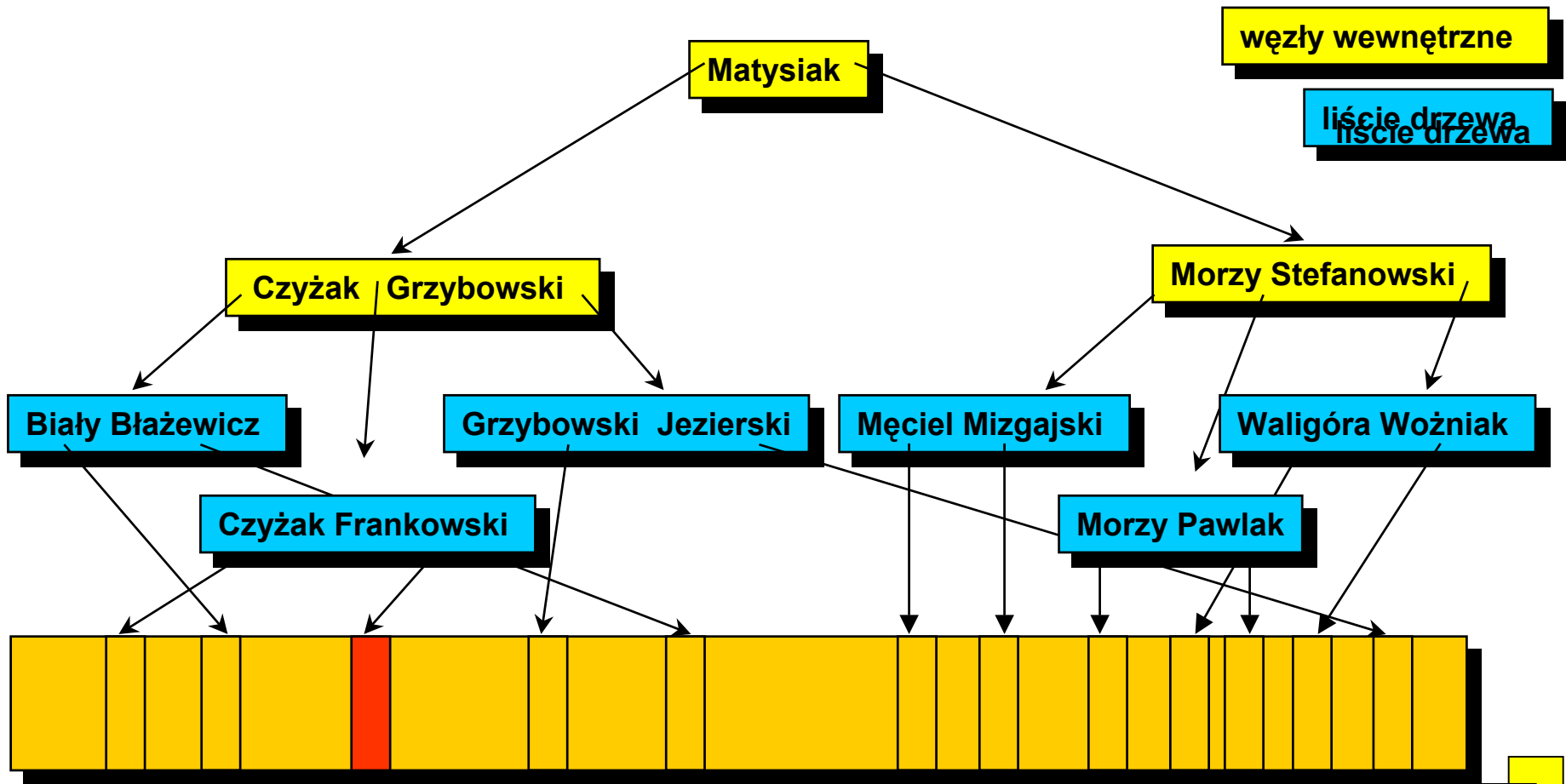
```
SELECT etat, AVG(placa_pod)  
FROM pracownicy GROUP BY etat;
```

- **UNIQUE:** eliminacja duplikatów

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

# Indeks B\*-drzewo

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



# Rodzaje B\*-drzew

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

```
CREATE [ UNIQUE ] INDEX nazwa ON tabela(atrybut1, atrybut2, ..)  
[ COMPRESS | NOCOMPRESS ] [ REVERSE ]  
[ COMPUTE STATISTICS ];
```

```
CREATE INDEX i_nazwisko ON pracownicy(nazwisko) COMPRESS;
```

```
CREATE UNIQUE INDEX i_complex  
ON pracownicy(id_prac, nazwisko, etat);
```

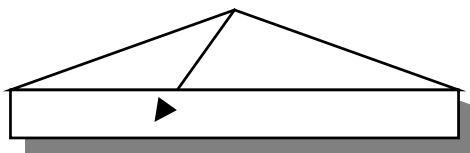


# Przesłanki do utworzenia indeksu B\*-drzewo

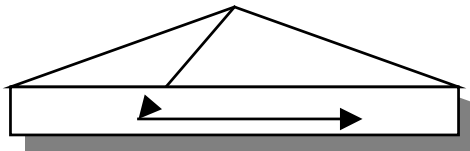
- na atrybutach często wykorzystywanych w warunkach selekcji,
- na atrybutach połączeniowych,
- tylko na atrybutach o dużej selektywności,
- na atrybutach rzadko modyfikowanych,
- na atrybutach będących kluczami obcymi (uniknięcie niepotrzebnego blokowania tabeli podrzędnej w przypadku operacji modyfikacji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

# Metody dostępu do indeksu B\*- drzewo

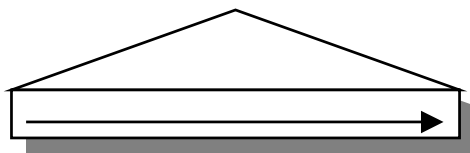
unikalne przeglądnięcie (ang. **unique scan**)



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

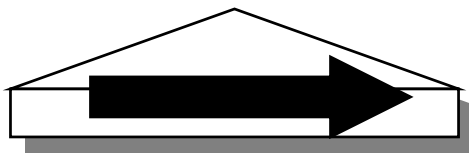


pełne przeglądnięcie (ang. **full scan**)



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

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



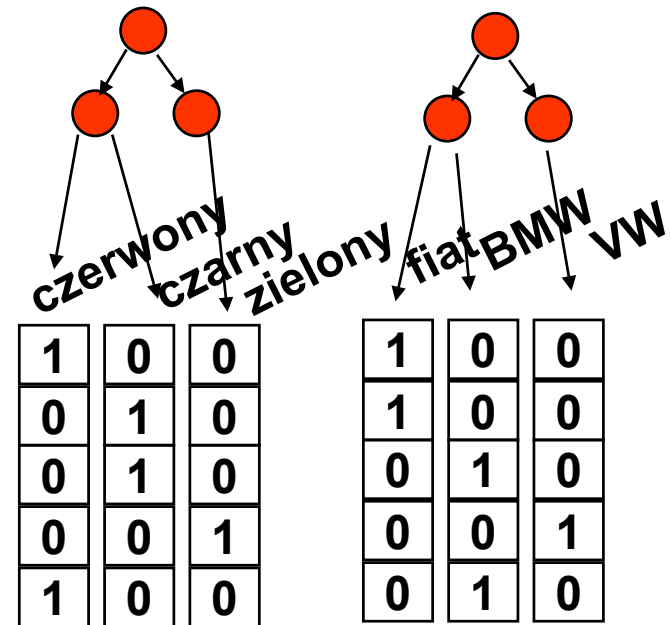
- odczyt wieloblokowy
- stosowany zamiast full table  
scan

# Indeks bitmapowy

PWG01425	czerwony	fiat
WAW3456	czarny	fiat
POZ3756	czarny	BMW
KTW3756	zielony	VW
PNR8956	czerwony	BMW

**SELECT count(\*) FROM samochody  
WHERE kolor IN  
( 'czerwony', 'zielony' )  
AND marka='fiat'**

$$\left( \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{array} \text{ OR } \begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{array} \right) \text{ AND } \begin{array}{c} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} = \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}$$



# Przesłanki do utworzenia indeksu bitmapowego

- w systemach przetwarzania analitycznego - OLAP,
- na atrybutach o małej selektywności,
- na atrybutach rzadko modyfikowanych
- dla zapytań z poszukiwaniem wartości pustych
- dla zapytań z dużą liczbą warunków OR i AND

```
CREATE BITMAP INDEX nazwa ON tabela (atrybut);
```

```
CREATE BITMAP INDEX b_etat ON pracownicy(etat)  
COMPUTE STATISTICS;
```

# Indeks B\*-drzewo vs. bitmapowy

## B\* - drzewo

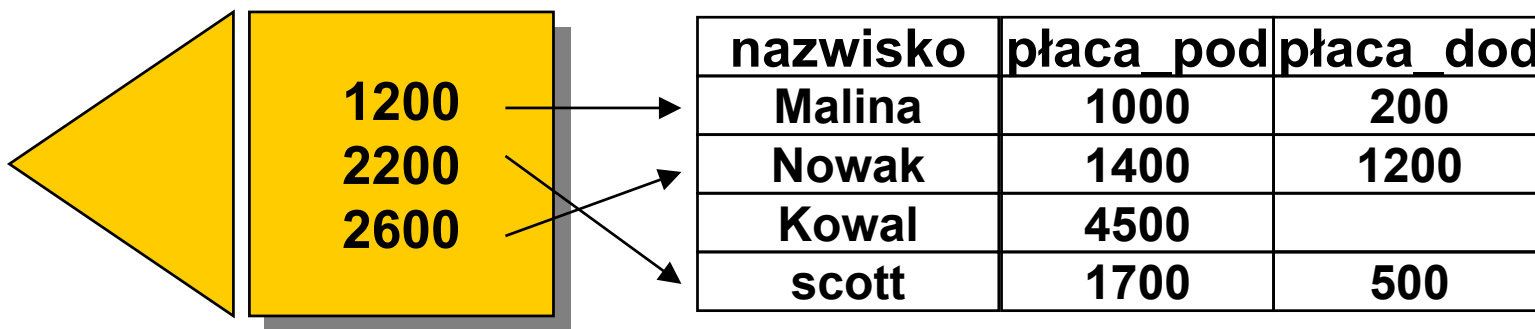
- skuteczny dla atrybutów z dużą dziedziną wartości
- efektywne wykonywanie operacji koniunkcji
- wielkość słabo zależna od wielkości dziedziny atrybutu
- bardzo wysoka współbieżność modyfikacji - blokada pojedynczego klucza indeksu
- niski koszt pojedynczej modyfikacji - modyfikacja pojedynczego klucza indeksu
- stosunkowo wysoki koszt modyfikacji grupy rekordów - każda wartość modyfikowana oddzielnie
- główne zastosowanie OLTP

## Bitmapowy

- skuteczny dla atrybutów z małą dziedziną wartości
- efektywne wykonywanie operacji alternatywy i koniunkcji
- wielkość bardzo silnie zależna od wielkości dziedziny atrybutu
- niska współbieżność modyfikacji - blokada całej bitmapy
- wysoki koszt pojedynczej modyfikacji - modyfikacji całej bitmapy (kompresja)
- stosunkowo niski koszt modyfikacji grupy rekordów - modyfikacja grupowa z możliwością zrównoleglenia
- główne zastosowanie OLAP

# Indeksy oparte na wyrażeniach

```
CREATE INDEX sum_płaca ON  
pracownicy (płaca_pod+płaca_dod);
```



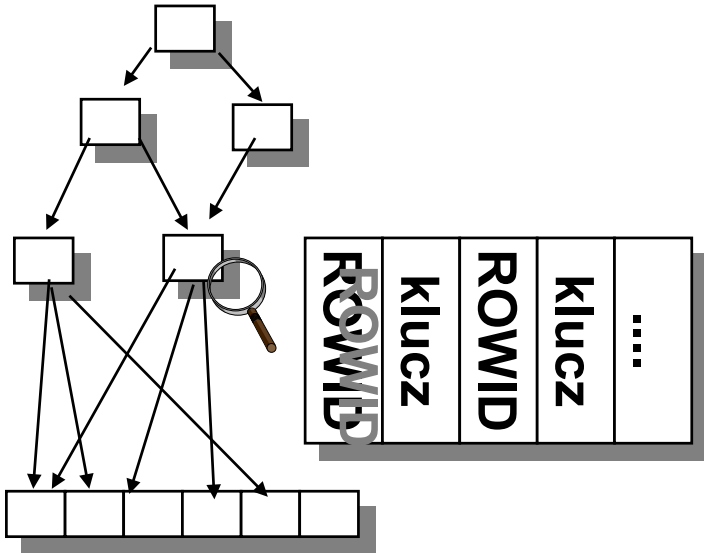
**Struktura fizyczna:**

- Bitmapowe
- B\*-drzewa

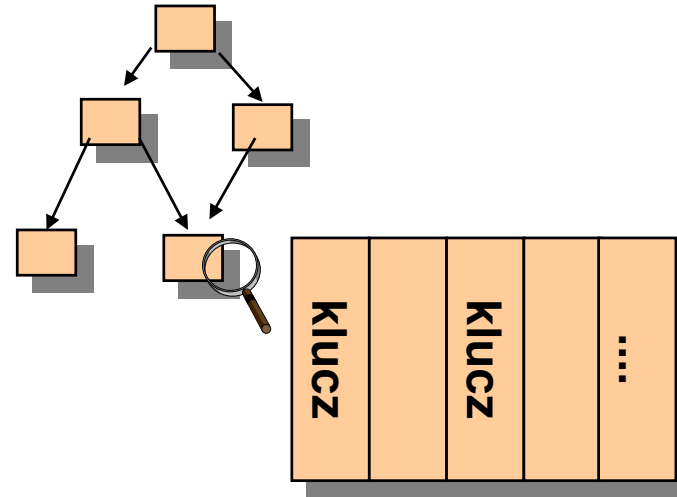
**przyśpieszają operacje (selekcja, połączenie) na wyrażeniach,  
mogą zawierać funkcje zdefiniowane przez użytkownika  
zalecane dla wyrażeń umieszczanych w klauzuli ORDER BY**

# Tabele zorganizowane jako indeks (ang. Index-Organized Table - IOT)

tabela z indeksem



IOT



- dla tabel przeglądanych zakresami wg klucza podstawowego
- dla tabel, które zawierają niewiele informacji oprócz klucza podstawowego
- dla tabel o nigdy nie modyfikowanym kluczu podstawowym
- w systemach przetwarzania analitycznego - OLAP

# Tworzenie IOT

```
CREATE TABLE nazwa (... , PRIMARY KEY (...))  
ORGANIZATION INDEX  
[PCTTHRESHOLD procent]  
[INCLUDING nazwa_kolumny]  
[OVERFLOW TABLESPACE nazwa_przestrzeni_tabel]
```

```
CREATE TABLE prac_zesp (  
id_prac NUMBER PRIMARY KEY,  
nazwisko VARCHAR2(20),  
nazwa VARCHAR2(20) )  
ORGANIZATION INDEX  
PCTTHRESHOLD 20  
OVERFLOW TABLESPACE USERS;
```



# Połączenia - nested loop

Odczyt  
jednokrotny



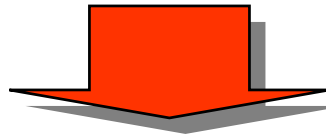
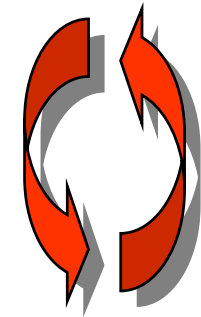
Tabela  
zewnętrzna

A	3
B	2
C	1
D	3

Tabela  
wewnętrzna

2	a
1	b
2	c
3	d
1	e

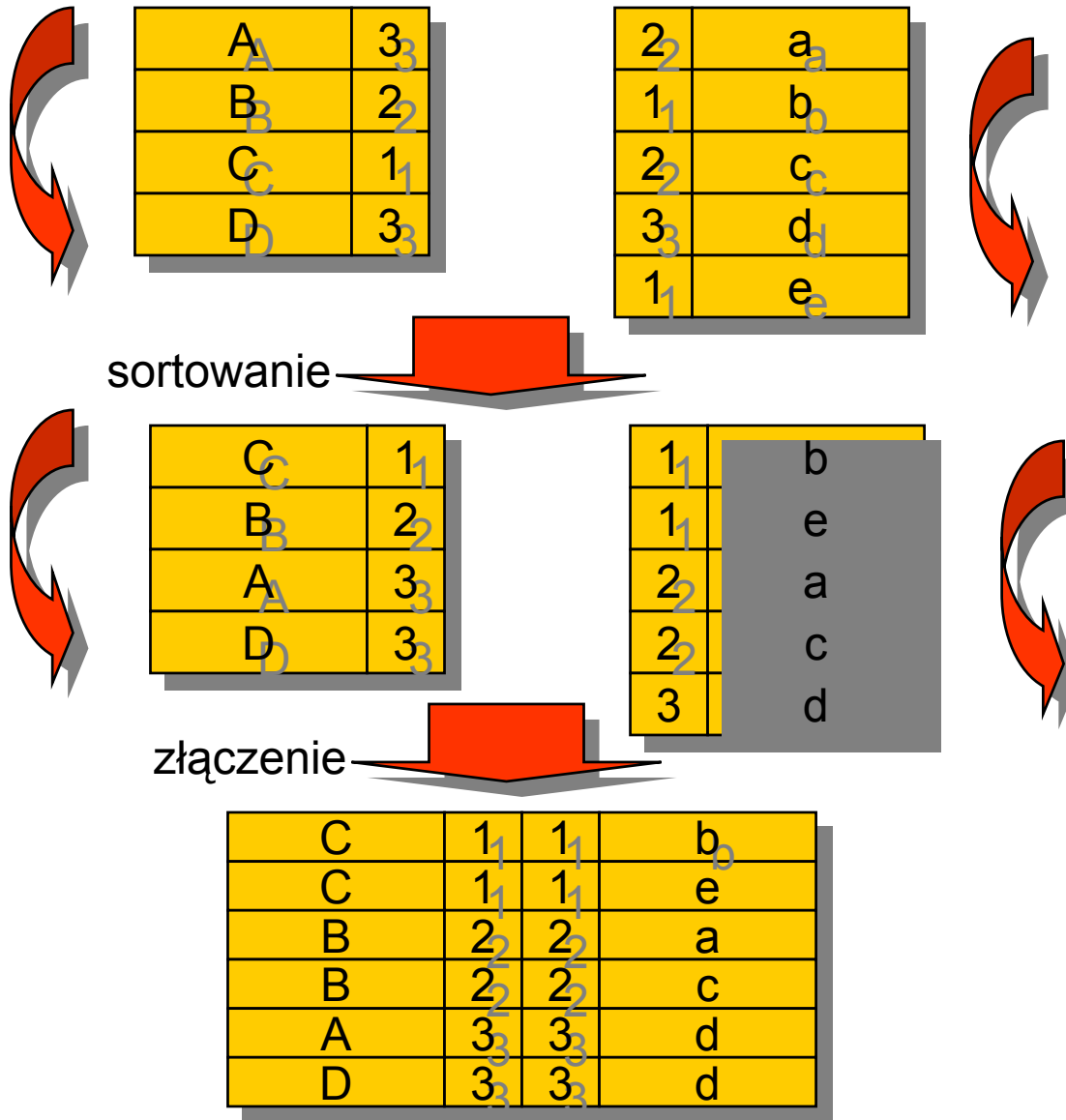
Odczyt  
wielokrotny



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

# Połączenia - sort merge



# Połączenia - hash join

Tabela  
zewnętrzna

A	3
B	2
C	1
D	3

Tabela  
wewnętrzna

2	a
1	b
2	c
3	d
1	e

Wartości funkcji  
haszującej

0	A	3
	D	3
1	C	1
2	B	2

Funkcja\_haszująca=  
kolumna\_połączeniowa mod 3

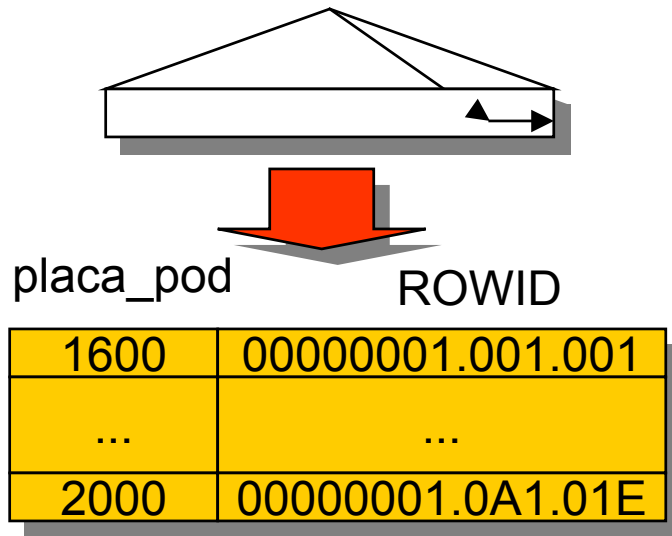
Wynik

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

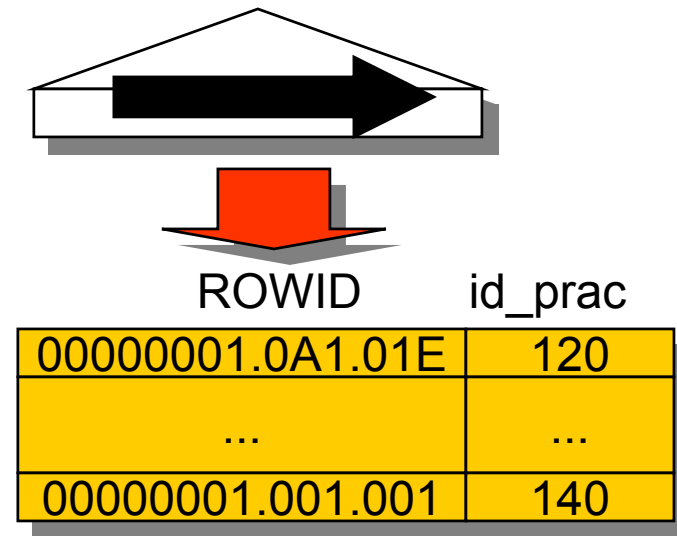
# Połączenia indeksów

```
SELECT id_prac  
FROM pracownicy  
WHERE placa_pod >1000;
```

*Range scan*(indeks na *placa\_pod*)



*Fast Full Scan*(indeksu na *id\_prac*)



join (hash)

120
...
140

# Wskazówki (ang. hints)

**Wskazówki umożliwiają określenie następujących elementów pracy optymalizatora:**

- rodzaj optymalizatora,
- cel optymalizacji,
- sposób dostępu do danych,
- kolejność łączonych tabel dla operacji połączenia,
- sposób realizacji połączenia

**Wskazówki umieszcza się w komentarzu bezpośrednio po klauzulach SELECT, INSERT, UPDATE, DELETE, przy czym pierwszym znakiem wskazówki musi być + (plus)**

```
SELECT /*+ CHOOSE */ nazwisko  
FROM pracownicy WHERE id_prac=100;
```

# Wybór optymalizatora

**CHOOSE** - kosztowy

**RULE** - regułowy

## Wybór celu optymalizacji

**ALL\_ROWS** - przepustowość

**FIRST\_ROWS** - czas odpowiedzi

**FIRST\_ROWS(n)** – czas odpowiedzi (pierwszych n krotek)

## Sposób dostępu do danych

**FULL\_SCAN**(nazwa\_tabeli) - pełne przeglądnięcie tabeli (ang. full scan)

**INDEX** (nazwa\_tabeli [nazwa\_indeksu]) - dostęp do rekordów za pomocą indeksu

**INDEX\_COMBINE** (nazwa\_tabeli [nazwa\_indeksu]) - dostęp do rekordów za pomocą indeksu bitmapowego

# Kolejność łączenia tabel

**ORDERED** - określenie kolejności łączenia tabel według kolejności określonej w klauzuli FROM

**STAR** - określenie kolejności łączenia tabel o schemacie gwiazdy z wykorzystaniem indeksu skonkatenowanego

## Sposób łączenia tabel

**USE\_NL**(*tabela\_wewnętrzna [...]*) - połączenie NESTED LOOP

**USE\_HASH** (*tabela\_wewnętrzna [...]*) - połączenie HASH JOIN

**USE\_MERGE** (*tabela\_wewnętrzna [...]*) - połączenie SORT MERGE

## Sposób dostępu do danych cd.

**INDEX\_ASC** (*nazwa\_tabeli [nazwa\_indeksu]*) - kierunek przeszukiwania indeksu przy selekcji zakresu wartości

**INDEX\_DESC** (*nazwa\_tabeli [nazwa\_indeksu]*) - kierunek przeszukiwania indeksu przy selekcji zakresu wartości

**INDEX\_FFS** (*nazwa\_tabeli [nazwa\_indeksu]*) - szybkie przeszukanie indeksu (ang. fast full scan)

**INDEX\_JOIN** (*nazwa\_tabeli [nazwa\_indeksu]*) - łączenie indeksów

**AND\_EQUAL**(*nazwa\_tabeli nazwa\_indeksu nazwa\_indeksu*) - powiązanie indeksów,

**USE\_CONCAT** - transformacja poleceń z operatorem OR na zapytania złożone z operatorem UNION ALL

**NO\_INDEX**(*nazwa\_tabeli [nazwa\_indeksu]*) - uniknięcie użycia indeksu

**NO\_EXPAND** - uniknięcie zamiany operatora IN na ciąg wyrażeń z operatorem OR

**MERGE\_SJ** - (umieszczona w podzapytaniu) - transformacja podzapytania z operatorem EXISTS na pół-złączenie

**HASH\_SJ** - (umieszczona w podzapytaniu) - transformacja podzapytania z operatorem EXISTS na pół-złączenie