

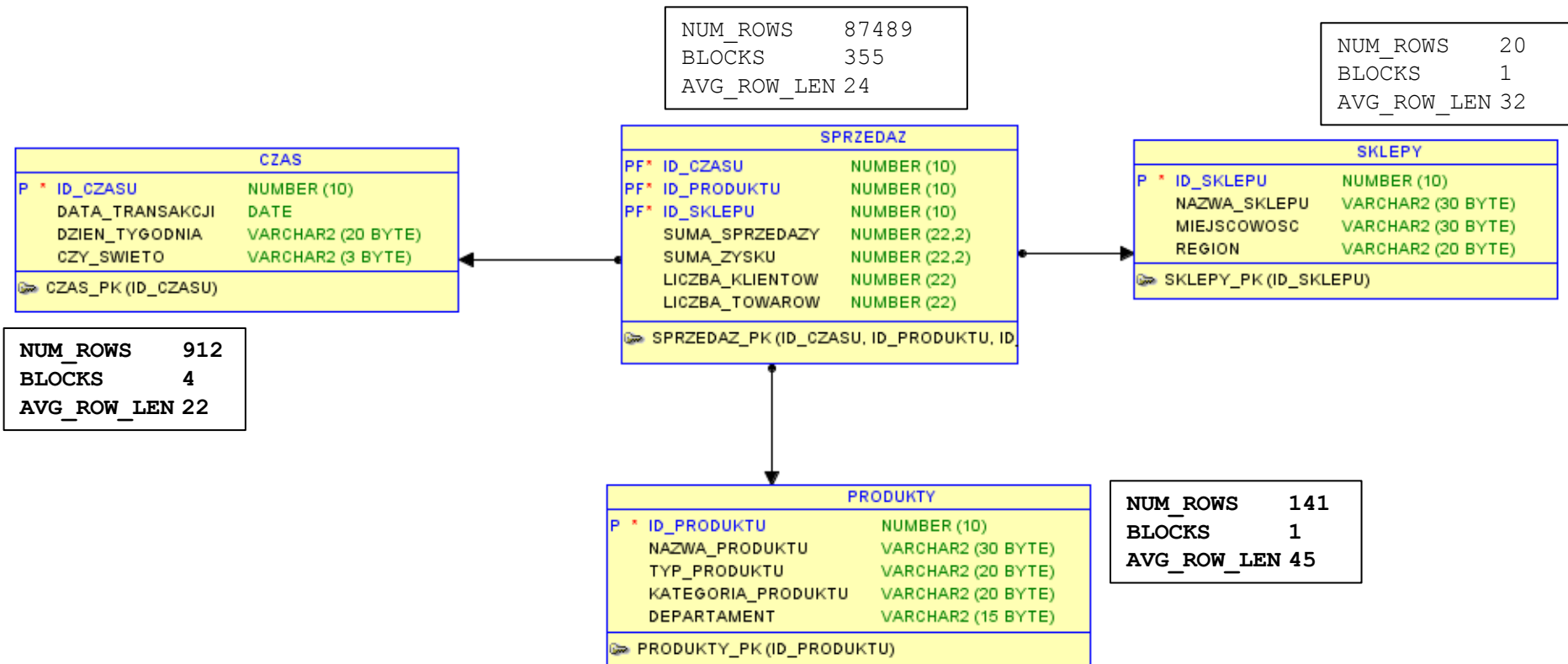
"Kilka słów" o strojeniu poleceń SQL
w kontekście Hurtowni Danych
wprowadzenie

Krzysztof Jankiewicz

Plan

- Opis schematu dla "kilku słów"
- Postać polecenia SQL
- Sposoby dostępu do tabel
- Indeksy B*-drzewo
- Indeksy binarne
- Zapytanie gwiazdziste
- Perspektywy materializowane
- Przepisywanie zapytań
 - Rola ograniczeń integralnościowych
 - Rola obiektów dimension

Opis schematu



Postać polecenia SQL

- Uzyskiwanie planu zapytania
- Trzy fazy przetwarzania polecenia SQL

```
select sum(sprzedaz.suma_sprzedazy) suma_sprzedazy,  
       sklepy.miejscowosc  
from   sprzedaz natural join sklepy natural join produkty  
where  sklepy.region = 'East'  
and    produkty.departament = 'Beverage'  
group by sklepy.miejscowosc
```

Elapsed: 00:00:00.040

```
select sum(sprzedaz.suma_sprzedazy) suma_sprzedazy,  
       sklepy.miejscowosc  
from   sprzedaz natural join sklepy  
where  sklepy.region = 'East'  
and    sprzedaz.id_produkту in (  
       select produkty.id_produkту  
       from   produkty  
       where  produkty.departament = 'Beverage')  
group by sklepy.miejscowosc
```

Elapsed: 00:00:00.034

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			106
HASH		GROUP BY	106
HASH JOIN			105
Access Predicates			
SPRZEDAZ.ID_SKLEPU=SKLEPY.ID_			
TABLE ACCESS	SKLEPY	FULL	2
Filter Predicates			
SKLEPY.REGION='East'			
HASH JOIN			103
Access Predicates			
SPRZEDAZ.ID_PRODUKTU=PRC			
TABLE ACCESS	PRODUKTY	FULL	2
Filter Predicates			
PRODUKTY.DEPARTAMENT			
TABLE ACCESS	SPRZEDAZ	FULL	99

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			106
HASH		GROUP BY	106
HASH JOIN			105
Access Predicates			
SPRZEDAZ.ID_SKLEPU=SKLEPY.ID_			
TABLE ACCESS	SKLEPY	FULL	2
Filter Predicates			
SKLEPY.REGION='East'			
HASH JOIN		RIGHT SEMI	103
Access Predicates			
SPRZEDAZ.ID_PRODUKTU=PRC			
TABLE ACCESS	PRODUKTY	FULL	2
Filter Predicates			
PRODUKTY.DEPARTAMENT			
TABLE ACCESS	SPRZEDAZ	FULL	99

Postać polecenia SQL

```
select sum(sprzedaz.suma_sprzedazy) suma_sprzedazy,
       sklepy.miejscowosc
from   sprzedaz natural join sklepy
where  sklepy.region = 'East'
and    exists (
        select 1
        from   produkty
        where  produkty.departament = 'Beverage'
        and    produkty.id_produktu = sprzedaz.id_produktu)
group by sklepy.miejscowosc
```

Elapsed: 00:00:00.035

```
select (select sum(sprzedaz.suma_sprzedazy)
        from   sprzedaz
        where  sprzedaz.id_sklepu = sklepy.id_sklepu
        and    exists (
                select 1
                from   produkty
                where  produkty.departament = 'Beverage'
                and    produkty.id_produktu =
                       sprzedaz.id_produktu)) suma_sprzedazy,
       sklepy.miejscowosc
from   sklepy
where  sklepy.region = 'East'
```

Elapsed: 00:00:00.056

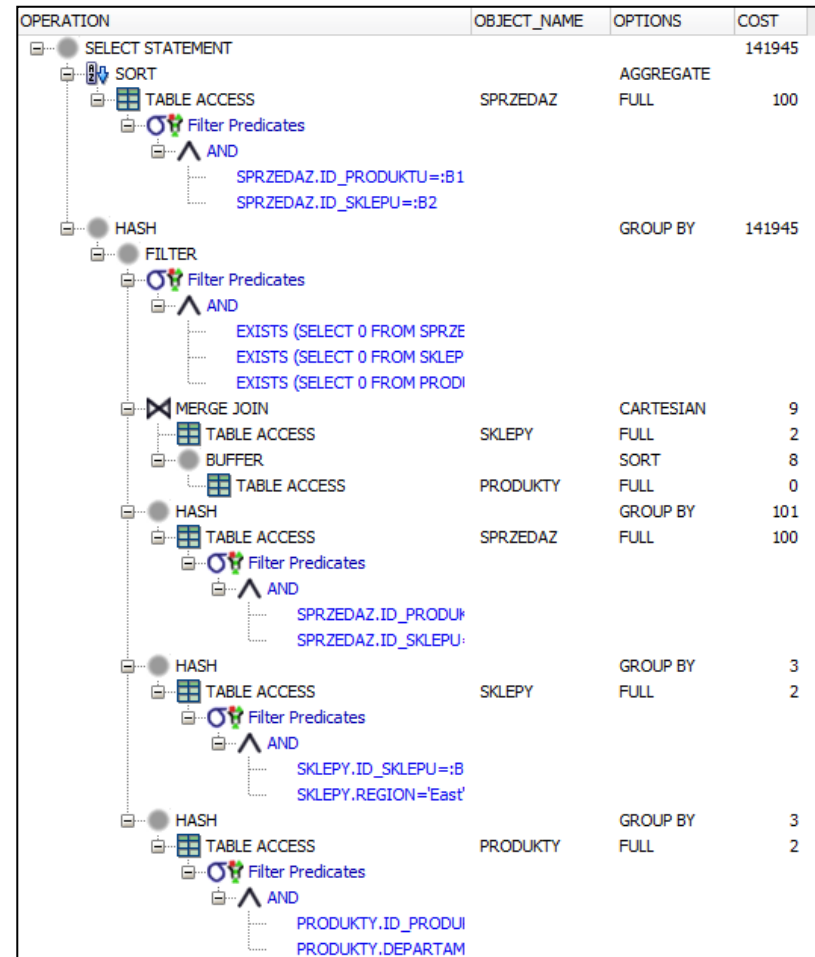
OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			106
HASH		GROUP BY	106
HASH JOIN			105
Access Predicates		SPRZEDAZ.ID_SKLEPU=SKLEPY.ID_	
TABLE ACCESS	SKLEPY	FULL	2
Filter Predicates		SKLEPY.REGION='East'	
HASH JOIN		RIGHT SEMI	103
Access Predicates		PRODUKTY.ID_PRODUKTU=SPRZ	
TABLE ACCESS	PRODUKTY	FULL	2
Filter Predicates		PRODUKTY.DEPARTAMENT	
TABLE ACCESS	SPRZEDAZ	FULL	99

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			2
SORT		AGGREGATE	
HASH JOIN		RIGHT SEMI	102
Access Predicates		PRODUKTY.ID_PRODUKTU=SPRZET	
TABLE ACCESS	PRODUKTY	FULL	2
Filter Predicates		PRODUKTY.DEPARTAMENT='Be	
TABLE ACCESS	SPRZEDAZ	FULL	100
Filter Predicates		SPRZEDAZ.ID_SKLEPU=:B1	
TABLE ACCESS	SKLEPY	FULL	2
Filter Predicates		SKLEPY.REGION='East'	

Postać polecenia

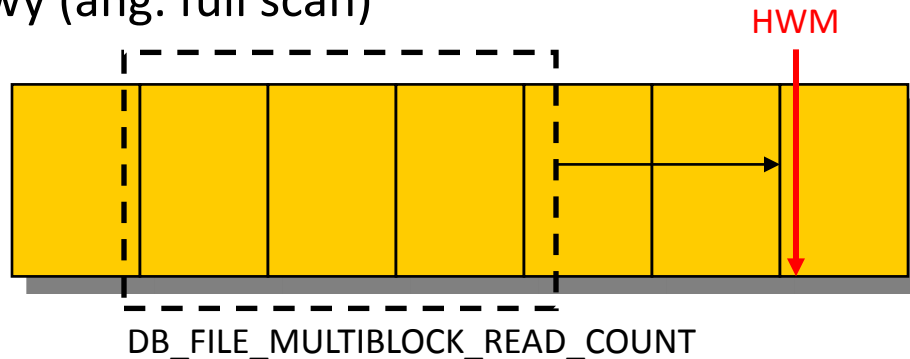
```
select sum(suma_sprzedazy), miejscowosc
from (
select (select sum(sprzedaz.suma_sprzedazy)
       from sprzedaz
       where sprzedaz.id_sklepu = sl.id_sklepu
       and   sprzedaz.id_produkту = pl.id_produkту) suma_sprzedazy,
sl.miejscowosc
from sklepy sl cross join produkty pl
where exists (
       select sum(sprzedaz.liczba_towarow)
       from sprzedaz
       where sprzedaz.id_sklepu = sl.id_sklepu
       and   sprzedaz.id_produkту = pl.id_produkту
       group by sprzedaz.suma_sprzedazy)
and   exists (
       select count(sklepy.nazwa_sklepu)
       from sklepy
       where sklepy.region = 'East'
       and   sklepy.id_sklepu = sl.id_sklepu
       group by sklepy.nazwa_sklepu)
and   exists (
       select sum(produkty.id_produkту)
       from produkty
       where produkty.departament = 'Beverage'
       and   produkty.id_produkту = pl.id_produkту
       group by produkty.kategoria_produkту)
group by miejscowosc
```

Elapsed: 00:00:18.175

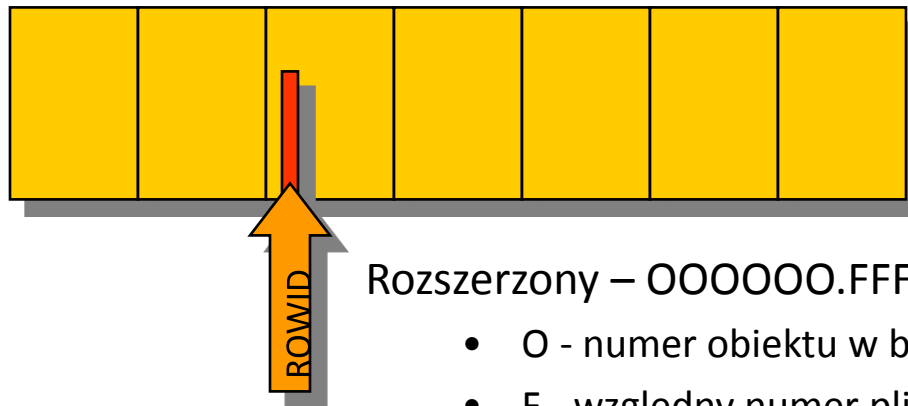


Metody dostępu do danych

Przeгляд liniowy (ang. full scan)



Dostęp za pomocą adresu rekordu (ang. ROWID)



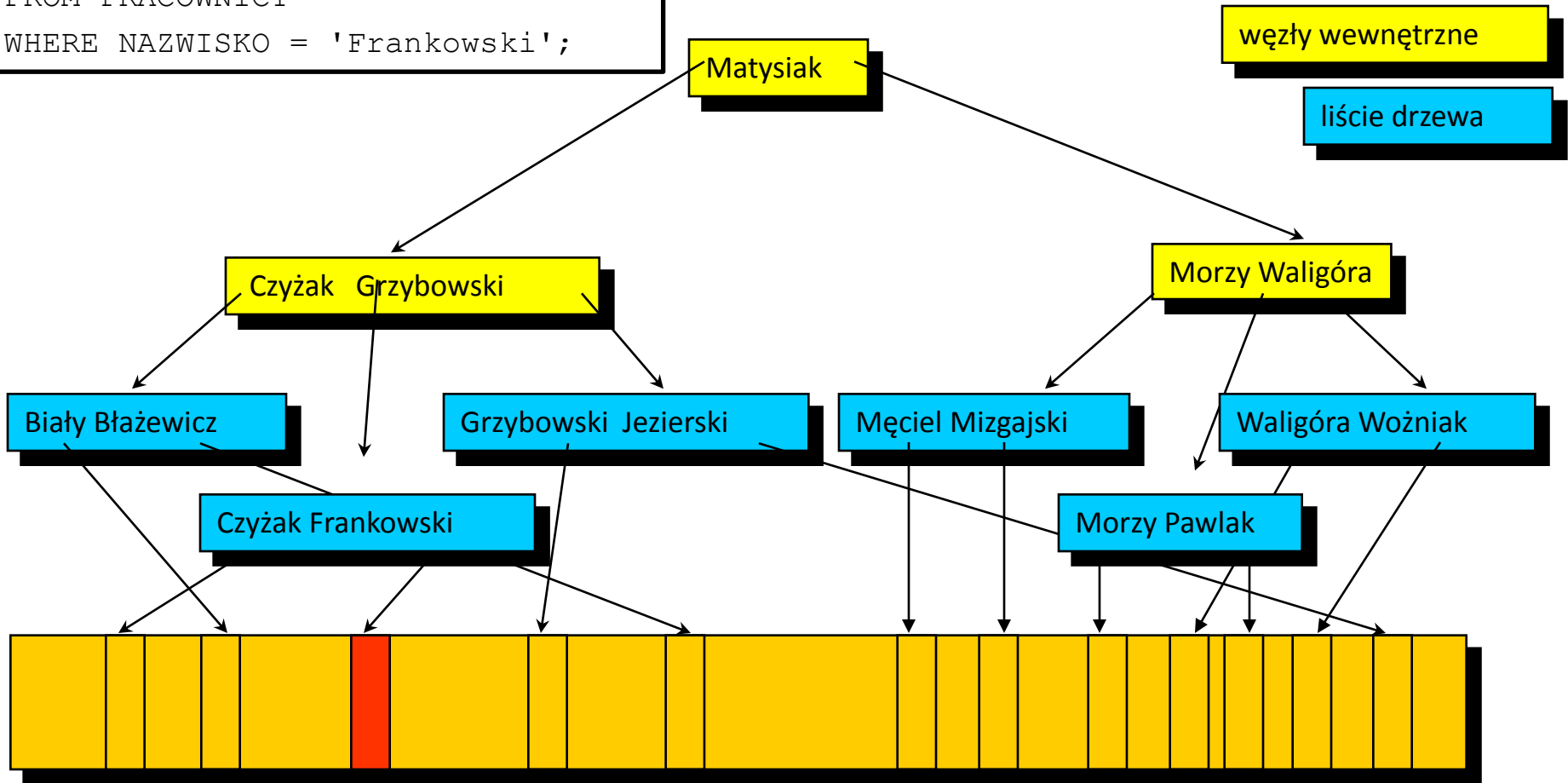
Rozszerzony – OOOOOO.FFF.BBBBBB.RRR

- 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

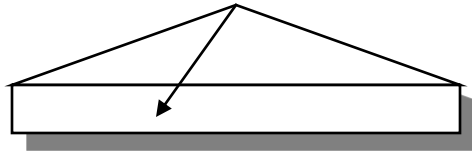
Indeks B*-drzewo

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

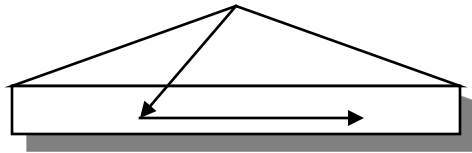


Metody dostępu do indeksu B*- drzewo

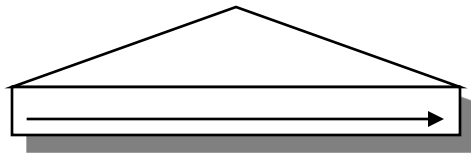
- unikalne przeglądnięcie (ang. **unique scan**)



- 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

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

B*-drzewo

```
select sum(sprzedaz.suma_sprzedazy) suma_sprzedazy,  
       sklepy.miejscowosc  
from   sprzedaz natural join sklepy natural join produkty  
where  sklepy.region = 'East'  
and    produkty.departament = 'Beverage'  
group by sklepy.miejscowosc
```

```
create index sklepy_region_idx  
on sklepy(region);
```

```
create index  
produkty_departament_idx on  
produkty(departament);
```

```
create index  
sprzedaz_id_produkту_idx on  
sprzedaz(id_produkту);
```

```
create index  
sprzedaz_id_sklepu_idx on  
sprzedaz(id_sklepu);
```

```
alter table sklepy add primary  
key(id_sklepu);
```

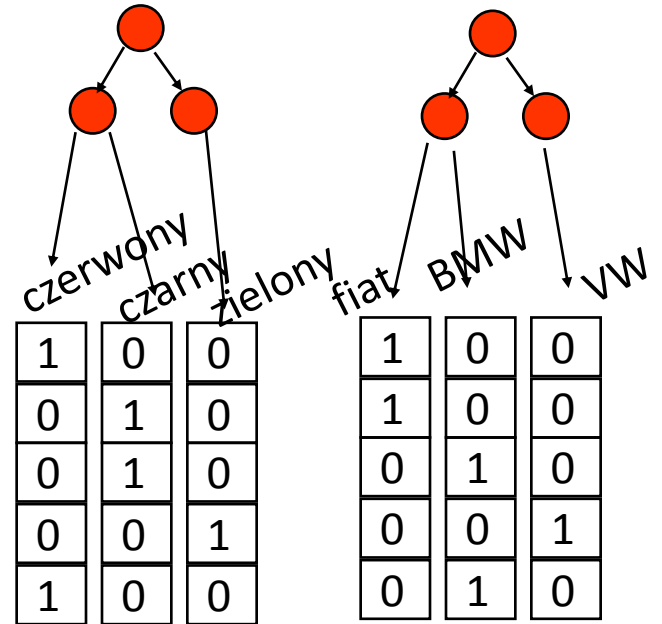
```
alter table produkty add primary  
key(id_produkту);
```

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			106
HASH		GROUP BY	106
HASH JOIN			105
Access Predicates			
AND			
		SPRZEDAZ.ID_PRODUKTU=PRC	
		SPRZEDAZ.ID_SKLEPU=SKLEPY	
MERGE JOIN		CARTESIAN	4
TABLE ACCESS	PRODUKTY	BY INDEX R...	2
INDEX	PRODUKTY_D...	RANGE SCAN	1
Access Predicates			
		PRODUKTY.DEPARTAM	
BUFFER		SORT	2
TABLE ACCESS	SKLEPY	BY INDEX R...	1
INDEX	SKLEPY_REGI...	RANGE SCAN	0
Access Predicates			
		SKLEPY.REGION='t	
TABLE ACCESS	SPRZEDAZ	FULL	99

Indeks bitmapowy

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

```
SELECT count(*) FROM samochody
WHERE kolor IN
      ('czzerwony', '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

Indeks bitmapowy

```
select count(sprzedaz.suma_sprzedazy) suma_sprzedazy,  
       sklepy.miejscowosc  
from   sprzedaz natural join sklepy natural join produkty  
where  sklepy.region = 'East'  
and    produkty.departament = 'Beverage'  
group by sklepy.miejscowosc
```

```
drop index  
sprzedaz_id_produkту_idx;  
  
drop index  
sprzedaz_id_sklepu_idx;  
  
create bitmap index  
sprzedaz_id_produkту_bidx on  
sprzedaz(id_produkту);  
  
create bitmap index  
sprzedaz_id_sklepu_bidx on  
sprzedaz(id_sklepu);
```

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			106
HASH		GROUP BY	106
HASH JOIN			105
Access Predicates			
AND			
SPRZEDAZ.ID_PRODUKTU=PRC			
SPRZEDAZ.ID_SKLEPU=SKLEPY			
MERGE JOIN		CARTESIAN	4
TABLE ACCESS	PRODUKTY	BY INDEX R...	2
INDEX	PRODUKTY_D...	RANGE SCAN	1
Access Predicates			
PRODUKTY.DEPARTAM			
BUFFER		SORT	2
TABLE ACCESS	SKLEPY	BY INDEX R...	1
INDEX	SKLEPY_REGI...	RANGE SCAN	0
Access Predicates			
SKLEPY.REGION='t			
TABLE ACCESS	SPRZEDAZ	FULL	99

Wskazówki i parametry inicjalizacyjne

- STAR_TRANSFORMATION_ENABLED
- STAR_TRANSFORMATION

```
select /*+ STAR_TRANSFORMATION */
      sum(sprzedaz.suma_sprzedazy) suma_sprzedazy,
      sklepy.miejscowosc
from   sprzedaz natural join sklepy natural join produkty
where  sklepy.region = 'East'
and    produkty.departament = 'Beverage'
group by sklepy.miejscowosc
```

