# Technologia HD w IBM DB2

**wykład przygotowany na podstawie materiałów IBM Polska**


**Robert Wrembel**
**Politechnika Poznańska**
**Instytut Informatyki**
`Robert.Wrembel@cs.put.poznan.pl`
`www.cs.put.poznan.pl/rwrembel`


## Technologie

- **InfoSphere Information Server ⇨ ETL**
- **Sfederowane bazy danych**
- **DB2**
  - **kompresja danych**
  - **indeks sklastrowany**
  - **Multidimensional Cluster**
  - **partycjonowanie**

# InfoSphere Information Server

- **InfoSphere Information Server ⇨ engine ETL**
- **Komponenty**
  - **InfoSphere Business Glossary**
  - **Industry Data Models**
  - **Metadata Workbench/Metadata Server**
  - **InfoSphere Information Analyzer**
  - **InfoSphere Quality Stage**
  - **InfoSphere Data Stage**
  - **InfoSphere Information Services Director**
  - **InfoSphere Federation Server**
  - **InfoSphere Change Data Capture**
  - **InfoSphere Information Server FastTrack**
- **Wdrożenia**
  - **Warta SA, PZU SA, BGŻ, Raiffeisen Bank Polska, Philipp Morris Int., Bakoma SA, TP SA, ...**



# InfoSphere Information Server

- **Business Glossary**
  - **definicje pojęć biznesowych**
  - **możliwość odwoływania się do tych pojęć z aplikacji**
- **Industry Data Models**
  - **predefiniowane modele HD m.in. dla zastosowań bankowych, ubezpieczeń, zarządzania finansami, służby zdrowia, handlu, telekomunikacji**
- **Metadata Workbench**
  - **analiza metadanych źródeł**
  - **analiza metadanych enginu ETL**
  - **change impact analysis**
  - **data provenance**

# InfoSphere Information Server

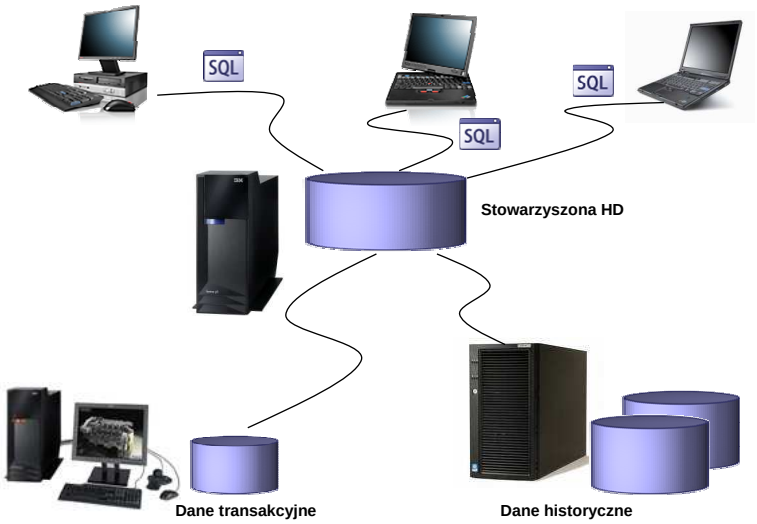
- Information Analyzer
  - profilowanie źródeł
  - analiza jakości danych
- Quality Stage
  - standaryzacja wartości
  - uzupełnianie wartości brakujących
  - eliminowanie duplikatów
  - predefiniowane techniki eliminowania duplikatów
- Data Stage
  - budowanie przepływów ETL (graficznie)
  - zarządzanie przepływami



# InfoSphere Information Server

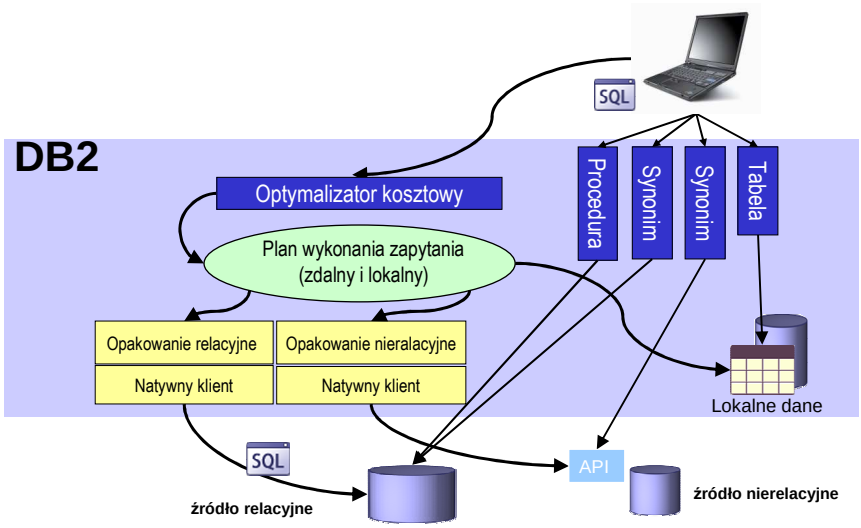
- FastTrack
  - wdrożenie/uruchomienie przepływów ETL
- Change Data Capture
  - wykrywanie zmian w danych źródłowych
  - propagowanie zmian do HD
  - wykrywanie zmian w strukturze źródeł
- Federation Server

# Stowarzyszone dane






# Stowarzyszone dane w DB2



# InfoSphere Federation Server

- Ujednolicenie interfejsów
- Zapis na wielu źródłach (2PC)
- Mapowanie użytkowników
- Wykonywanie zdalnych procedur
- Optymalizacja dostępu
  - parametry opisujące źródło (wydajność CPU, systemu dyskowego, przepustowość sieci, ...)
  - statystyki optymalizacyjne dla zdalnych obiektów
- Agregaty oparte na zdalnych źródłach



# Kompresja

- Wartości puste
  - nie zajmują miejsca (wer. 8)
- Kompresja rekordów (wer. 9)
- Multidimensional Clustering (wer. 8)

# Kompresja rekordów (1)

- Kompresja słownikowa
  - algorytm kompresji bazuje na alg. Lempel-Ziv (LZ)
  - statyczny słownik budowany w czasie reorganizacji tabeli
  - rozmiar ~100KB
- Dane są przechowywane w postaci skompresowanej na dysku i w buforze danych
  - zmniejszenie operacji I/O
  - zmniejszenie zajętości bufora danych
  - konieczność dekompresowania danych przed wykonaniem operacji (koszt CPU)



# Kompresja rekordów (2)

- Powtarzające się ciągi symboli są zastępowane pozycją słownikową o znacznie mniejszej długości
- Konieczność utrzymywania słownika kompresji

| Kosmetyk | Data | Sklep | Kwota |
|---|---|---|---|
| Polo Ralph Laurent | 20-05-06 | Sefora | 160 |
| Eternity Calvin Klein | 21-05-06 | Sefora | 300 |
| Polo Ralph Laurent | 22-05-06 | Douglas | 170 |
| Eternity Calvin Klein | 23-05-06 | Sefora | 165 |
| Eternity Calvin Klein | 23-06-06 | Douglas | 180 |

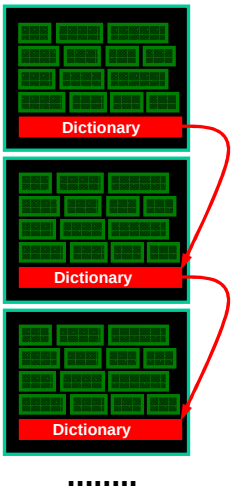
**słownik kompresji**

| Wartość | Kod |
|---|---|
| Polo Ralph Laurent | 01 |
| Eternity Calvin Klein | 02 |
| Sefora | 03 |
| Douglas | 04 |

| Kosmetyk | Data | Sklep | Kwota |
|---|---|---|---|
| 01 | 20-05-06 | 03 | 160 |
| 02 | 21-05-06 | 03 | 300 |
| 01 | 22-05-06 | 04 | 170 |
| 02 | 23-05-06 | 03 | 165 |
| 02 | 23-06-06 | 04 | 180 |

# Kompresja rekordów (6)

- Słownik kompresji składowany w
  - osobnej tabeli
  - w bloku razem z danymi
    - słownik kompresji nie jest widziany przez użytkownika






# Kompresja rekordów (3)

- CREATE TABLE <table name> --->

```
          |---COMPRESS NO---|
       ----+-----------------+---->
          |---COMPRESS YES--|
```

- ALTER TABLE <table name> --->

```
         --+----------------------+---->
           |--COMPRESS--+-YES--+--|
                        |--NO--|
```

# Kompresja rekordów (4)

- Przed kompresją jest budowany słownik kompresji na podstawie
  - zawartości całej tabeli
  - próbki danych wpisanych do tabeli

```
>--REORG--<table name>--+----------------------+---->
                        '--INDEX--<index name>--'
     .-ALLOW READ ACCESS-.
>--+-+-------------------+--+---------------+--+-----------+-->
     '-ALLOW NO ACCESS---'  '-USE--<tbspace>-'  '-INDEXSCAN-'
                   .-KEEPDICTIONARY---.
>--+-------------+-+------------------+-+-->
   '-LONGLOBDATA-' '-RESETDICTIONARY---'
```



# Kompresja rekordów (5)

**RESETDICTIONARY**

| tabela kompresowana | słownik istnieje | wynik |
|---|---|---|
| tak | tak | nowy słownik; kompresja |
| tak | nie | nowy słownik; kompresja |
| nie | tak | słownik usunięty; brak kompresji |
| nie | nie | nie wpływa |

**KEEPDICTIONARY**

| tabela skompresowana | słownik istnieje | wynik |
|---|---|---|
| tak | tak | stary słownik pozostaje; kompresja |
| tak | nie | nowy słownik; kompresja |
| nie | tak | stary słownik pozostaje; brak kompresji |
| nie | nie | nie wpływa |

# Kompresja rekordów (7)

- Scenariusz 1 kompresji
  - utworzyć tabelę z kompresją
    - `CREATE TABLE Sales COMPRESS YES`
  - wczytać reprezentatywną próbkę danych
    - `LOAD FROM filesmall OF DEL REPLACE INTO Sales`
  - utworzyć słownik kompresji na podstawie próbki
    - `REORG TABLE Sales`
  - wczytać docelowe dane do tabeli
    - `LOAD FROM filerest OF DEL INSERT INTO Sales`



# Kompresja rekordów (8)

- Scenariusz 2 kompresji
  - utworzyć tabelę z kompresją
    - `ALTER TABLE Sales COMPRESS YES`
    - `CREATE TABLE NewSales … COMPRESS YES`
  - wczytać dane do tabeli
    - `LOAD FROM file OF DEL REPLACE INTO NewSales`
  - zbudować słownik kompresji i skompresować dane
    - `REORG NewSales …`
  - dołączyć skompresowaną tabelę jako partycję innej tabeli
    - 
```
ALTER TABLE Sales
    ATTACH PARTITION Mar05
    STARTING '03/01/2005'
    ENDING '03/31/2005'
    FROM TABLE NewSales
```

# Jakość kompresji (1)

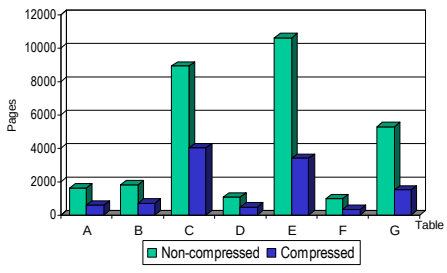
| | | | No Compression | | Row Compression | | |
|---|---|---|---|---|---|---|---|
| TPCH Table | Row Count | Page Size | Page Count | Rows/Page | Page Count | Rows/Page | Ratio |
| region | 5 | 8k | 2 | 3 | 1 | 5 | 50% |
| | | 16k | 2 | 3 | 1 | 5 | 50% |
| nation | 25 | 8k | 2 | 13 | 1 | 25 | 50% |
| | | 16k | 2 | 13 | 1 | 25 | 50% |
| supplier | 100,000 | 8k | 2008 | 50 | 1064 | 94 | 47% |
| | | 16k | 995 | 101 | 530 | 189 | 47% |
| customer | 1,500,000 | 8k | 33877 | 44 | 18219 | 82 | 46% |
| | | 16k | 16770 | 89 | 9043 | 166 | 46% |
| part | 2,000,000 | 8k | 37285 | 54 | 11503 | 174 | 69% |
| | | 16k | 18486 | 108 | 7875 | 254 | 57% |
| partsupp | 8,000,000 | 8k | 156576 | 51 | 49872 | 160 | 68% |
| | | 16k | 77546 | 103 | 31581 | 253 | 59% |
| orders | 15,000,000 | 8k | 218358 | 69 | 86523 | 173 | 60% |
| | | 16k | 108309 | 138 | 59063 | 254 | 45% |
| lineitem | 59,986,052 | 8k | 1023162 | 59 | 426292 | 141 | 58% |

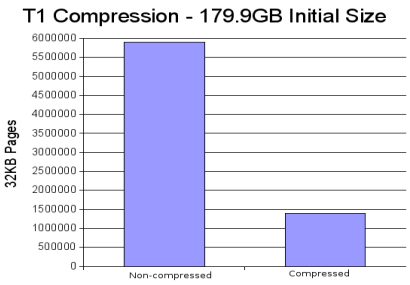


# Jakość kompresji (2)

| Compression Type | 32KB Page Count | Disk space |
|---|---|---|
| No compression | 5893888 | 179.9GB |
| Row compression | 1392446 | 42.5GB |

% Pages Saved: 76.4%



# Indeks sklastrowany (1)

- Indeks sklastrowany (ang. clustering index) kontroluje porządek składowania rekordów na dysku
- Po zdefiniowaniu indeksu wstawiane rekordy są składowane w kolejności kluczy indeksowych (z liści indeksu)
- Reorganizacja tabeli powoduje reorganizację rekordów na dysku, zgodnie z kolejnością kluczy indeksowych
- Dla danej tabeli jest możliwy tylko jeden indeks sklastrowany (jeden porządek fizycznego ułożenia rekordów)



# Indeks sklastrowany (2)

Indeks sklastrowany

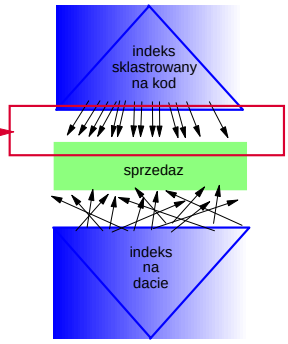
Blok danych 1

| kod | region | data | ilosc | wart |
|---|---|---|---|---|
| 101 | 21 | 04/02 | 1 | 1.50 |
| 101 | 21 | 04/02 | 1 | 1.50 |
| 101 | 7 | 04/02 | 2 | 3.00 |
| 101 | 7 | 04/01 | 6 | 8.11 |

Blok danych 2

| kod | region | data | ilosc | wart |
|---|---|---|---|---|
| 101 | 7 | 04/02 | 1 | 1.50 |
| 101 | 21 | 04/02 | 3 | 4.10 |
| 101 | 7 | 04/01 | 2 | 3.00 |

Klastrowanie




```
CREATE INDEX kod_idx ON SPRZEDAZ(kod)
CLUSTER
```

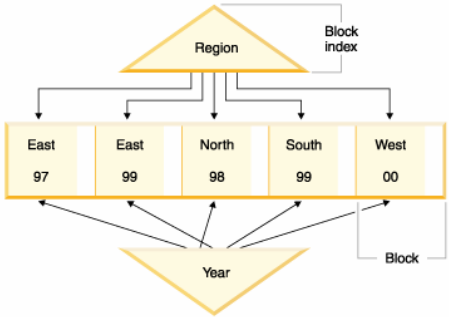
# Indeks sklastrowany (3)

- Zastosowanie do zapytań z:
  - group by
  - order by
  - distinct
- Indeks sklastrowany eliminuje sortowanie



# MDC (1)

- Klastrowanie wielowymiarowe (MultiDimensional Clustering - MDC)
  - tzw. indeks blokowy (ang. block index) - wskazuje na blok danych
  - grupowanie danych na podstawie wartości wielu wymiarów



# MDC (2)

Tabela bez MDC

Blok danych 1

| kod | region | data | ilosc | wart |
|---|---|---|---|---|
| 101 | 21 | 04/02 | 1 | 1.50 |
| 101 | 21 | 04/02 | 1 | 1.50 |
| 101 | 7 | 04/02 | 2 | 3.00 |
| 101 | 7 | 04/01 | 6 | 8.11 |

Blok danych 2

| kod | region | data | ilosc | wart |
|---|---|---|---|---|
| 101 | 7 | 04/02 | 1 | 1.50 |
| 101 | 21 | 04/02 | 3 | 4.10 |
| 101 | 7 | 04/01 | 2 | 3.00 |

Tabela z MDC (region, data)

Blok danych 1

| kod | region | data | ilosc | wart |
|---|---|---|---|---|
| 101 | 21 | 04/02 | 1 | 1.50 |
| 101 | 21 | 04/02 | 1 | 1.50 |
| 101 | 21 | 04/02 | 3 | 4.10 |

Blok danych 2

| kod | region | data | ilosc | wart |
|---|---|---|---|---|
| 101 | 7 | 04/01 | 6 | 1.50 |
| 101 | 7 | 04/01 | 2 | 3.00 |

Blok danych 3

| kod | region | data | ilosc | wart |
|---|---|---|---|---|
| 101 | 7 | 04/02 | 2 | 3.00 |
| 101 | 7 | 04/02 | 1 | 1.50 |

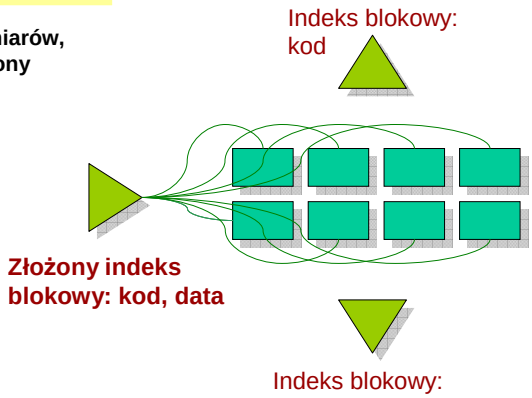
```
CREATE TABLE sprzedaz
( kod  BIGINT, region INT, data DATE,
  ilosc INT, wartosc DECIMAL(16,2))
ORGANIZE BY (region, data);
```



# MDC (3)

```
ORGANIZE BY (kod, data);
```

**Oprócz indeksów dla wymiarów, tworzony jest indeks złożony**

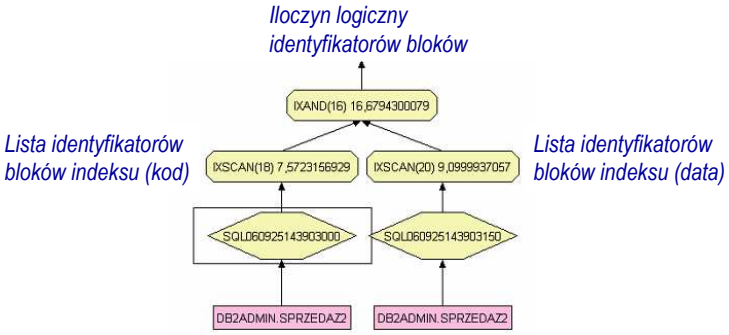


# MDC - optymalizacja zapytań

➲ **Część wspólna indeksów blokowych**

```
select sum(wartosc), kod, data
from sprzedaz
where kod=20 and data='20-01-2006'
group by kod, data
```

**Technika stosowana również do indeksów zwykłych i blokowych**






# MDC – podsumowanie

Poprawa wydajności odczytów, ponieważ dane są fizycznie poukładane są na dysku (mniejsza liczba operacji we/wy)

Poprawa wydajności modyfikacji danych w porównaniu do tradycyjnych indeksów. Wstawienie rekordu do istniejącego bloku nie wymaga modyfikacji indeksu (indeks wskazuje do bloku, a nie do rekordu)

Zagwarantowane sklastrowanie danych. Reorganizacja danych nie jest wymagana

# Partycjonowanie tabel (1)

```
CREATE TABLE sprzedaz
(
  kod  BIGINT,
  region INT,
  data DATE,
  ilosc INT,
  wartosc DECIMAL(16,2)
)
ORGANIZE BY (kod, region)
PARTITION BY RANGE (data)
(
  PARTITION part1 STARTING FROM MINVALUE IN dane1,
  PARTITION part2 STARTING FROM '2006-03-31' IN dane2,
  PARTITION part3 STARTING FROM '2006-07-01' IN dane3,
  PARTITION part4 STARTING FROM '2006-10-01'
                                        ENDING '2006-12-31' IN dane4
)
```

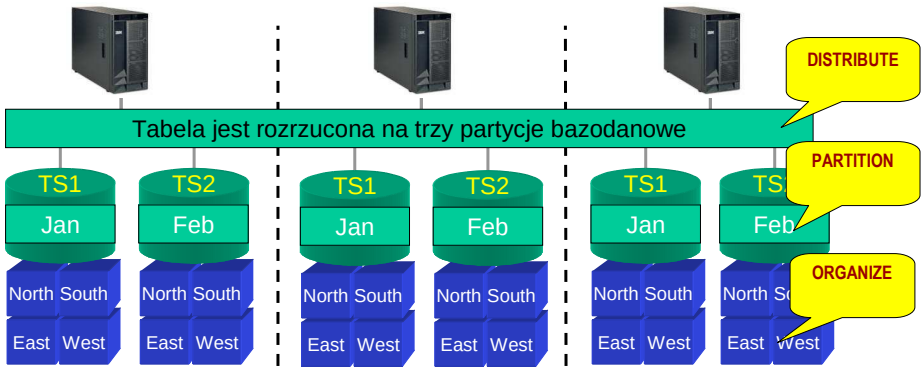


# Partycjonowanie BD na wiele komputerów

```
CREATE TABLE sprzedaz
(
  id_trans BIGINT,
  kod  BIGINT,
  region INT,
  data DATE,
  ilosc INT,
  wartosc DECIMAL(16,2)
)
DISTRIBUTE BY HASH (id_trans)
ORGANIZE BY (kod, region)
PARTITION BY RANGE (data)
(
  PARTITION part1 STARTING FROM MINVALUE IN dane1,
  PARTITION part2 STARTING FROM '2006-03-31' IN dane2,
  PARTITION part3 STARTING FROM '2006-07-01' IN dane3,
  PARTITION part4 STARTING FROM '2006-10-01'
                                        ENDING '2006-12-31' IN dane4
)
```

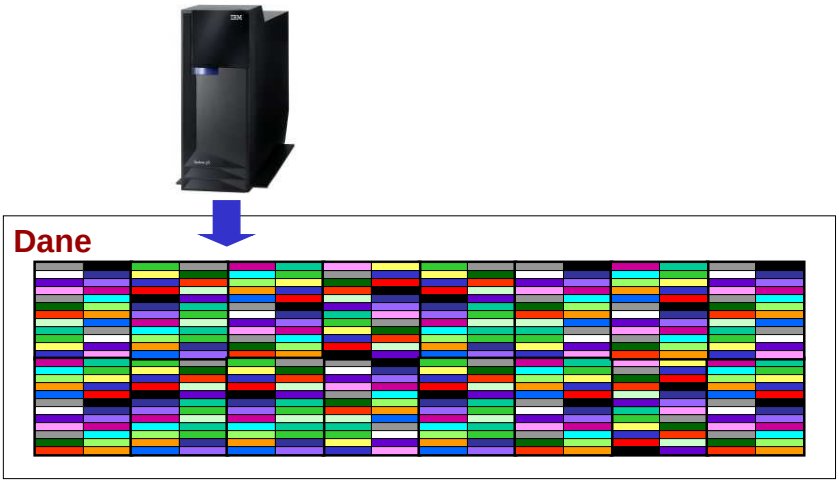
# Połączenie mechanizmów partycjonowania

- DISTRIBUTE BY HASH
- PARTITION BY RANGE
- ORGANIZE BY DIMENSIONS

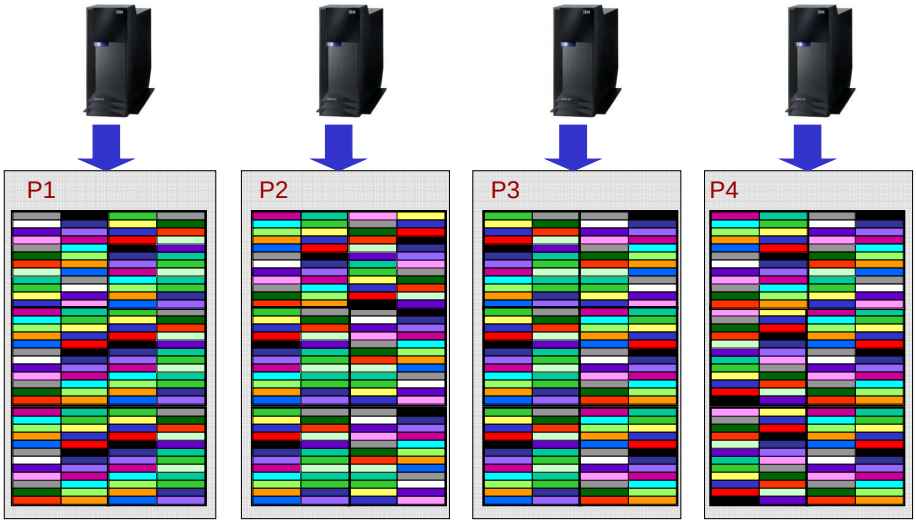





# Bez partycjonowania

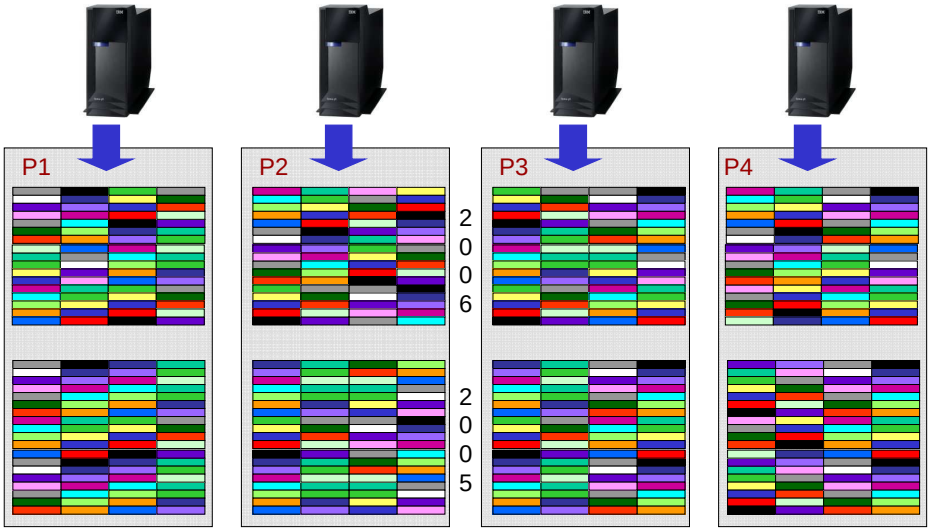


# Partycjonowanie na wielu komputerach

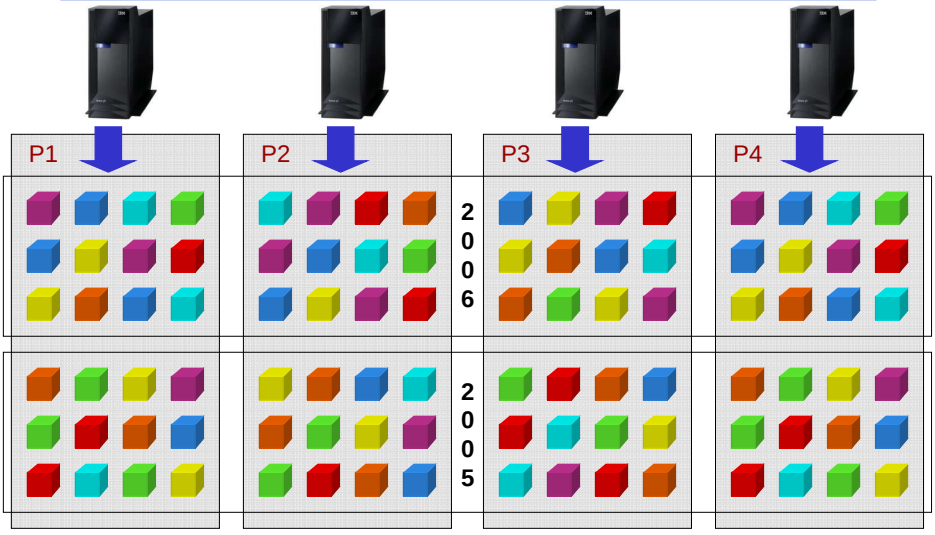





# + partycjonowanie tabel



# + MDC






# Studium przypadku

- Operator komórkowy w USA
- Charakterystyka
  - 10 miliardów transakcji na dzień
  - 32 TB surowych danych
  - Tysiące współbieżnych użytkowników
  - 7,000 użytkowników obsługi klienta
  - do 37000 zapytań/dzień
  - Hurtownia danych ładowana na bieżąco
  - Doładowanie ponad 1 miliarda zdarzeń dziennie (do 1.6 miliarda)
  - DB2 DWE, SAS, 16 x 8 CPU P5 pSeries
- MDC
  - szybsze o 80% usuwanie danych
  - o 43% mniej odwołań I/O