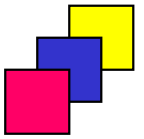
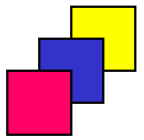
Optymalizacja poleceń SQL

Połączenia



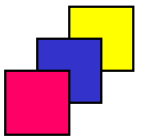
Połączenie relacji (1)

- **Operacja binarna – zawsze udział biorą dwie tabele, jedna zostaje nazwana tabelą zewnętrzną, druga tabelą wewnętrzną.**
- **W przypadku polecenia łączącego więcej niż dwie tabele (np. A , B i C), połączenie realizowane jest zawsze dla pary tabel (np. A z B, wynik z C, albo A z C i wynik z B, itd.).**
- **Podstawowe zasady:**
  - **główna zasada: kolejność łączenia tabel powinna jak najbardziej ograniczać zbiór rekordów**
  - **optymalizator szuka w zbiorze łączonych tabel takich, których połączenie wyprodukuje 1 rekord – jeśli znajdzie, te tabele są łączone na początku**
  - **w przypadku połączenia zewnętrznego tabela zewnętrzna jest umieszczana w kolejności za tabelą wewnętrzną**



Połączenie relacji (2)

- **Realizowane przy użyciu jednego z algorytmów:**
  - **nested loops,**
  - **sort merge,**
  - **hash join.**
- **Wybór algorytmu zależy od:**
  - **rozmiaru tabeli,**
  - **postaci warunku połączeniowego,**
  - **spodziewanego rozmiaru wyniku połączenia,**
  - **dostępności i rozmiaru obszaru sortowania,**
  - **wartości parametru odczytu wieloblokowego (DB_FILE_MULTIBLOCK_READ_COUNT).**



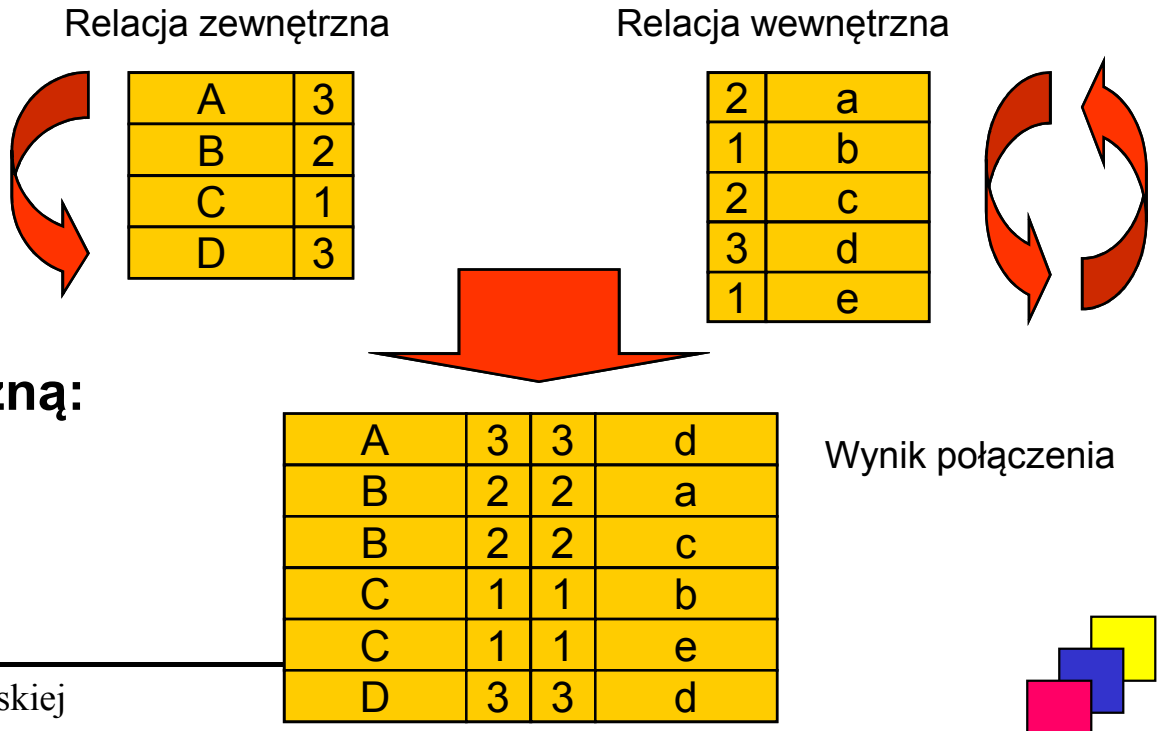
Algorytm nested loops

- **Stosowany, gdy:**
  - **w połączeniu bierze udział mała część rekordów relacji,**
  - **istnieje efektywna metoda dostępu do danych relacji wewnętrznej (indeks założony na kolumnie w warunku połączeniowym).**
- **Główny koszt – koszt odczytu rekordów relacji zewnętrznej i znalezienia odpowiadających rekordów relacji wewnętrznej.**
- **Algorytm:**
- **W planie wykonania relacja zewnętrzna ponad relacją wewnętrzną:**

  NESTED LOOPS

  relacja_zewnętrzna

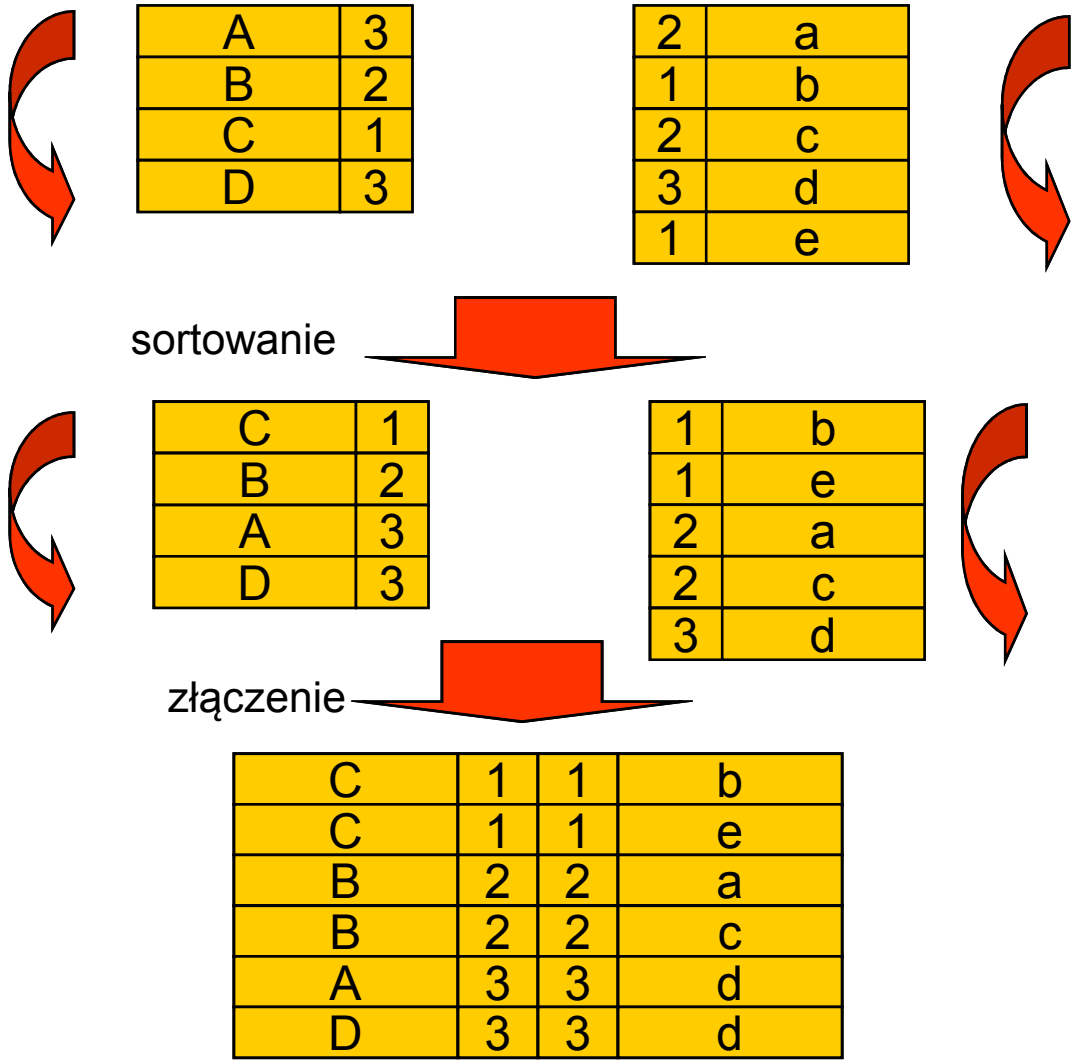
  relacja_wewnętrzna



Algorytm sort merge (1)

- **Stosowany, gdy:**
  - **łączone relacje są niezależne (brak połączenia kluczem obcym),**
  - **warunek połączeniowy z operatorami: <, <=, >, >= (ale nie !=) i duże rozmiary łączonych relacji (zachowuje się lepiej niż nested loop),**
  - **relacja już są posortowane lub nie ma potrzeby realizacji sortowania (bo np. istnieje odpowiedni indeks).**
- **Główny koszt – koszt wczytania obu relacji do pamięci i ich posortowania.**
- **Brak podziału na relację zewnętrzną i wewnętrzną.**
- **Algorytm:**
  1. **Posortowanie obu relacji ze względu na wartości kolumn w warunku połączeniowym.**
  2. **Połączenie rekordów o tych samych wartościach kolumn w warunku połączeniowym.**

Algorytm sort merge (2)



Algorytm hash join (1)

- **Stosowany, gdy:**
  - **warunek połączeniowy jest warunkiem równościowym, i**
  - **łączone relacje o dużym rozmiarze lub większa część rekordów mniejszej relacji bierze udział w połączeniu.**
- **Główny koszt – zbudowanie tabeli haszowej dla relacji zewnętrznej i odczyt rekordów z relacji wewnętrznej.**
- **Relacja zewnętrzna – mniejsza z relacji, najlepiej, jeśli mieści się w pamięci.**
- **W planie wykonania pierwsza relacja, z której zbudowano tablicę haszową:**

```
HASH JOIN
    relacja_zewnętrzna
    relacja_wewnętrzna
```

Algorytm hash join (2)

- **Algorytm:**