

Indeksy w SZBD

Robert A. Kłopotek

r.klopotek@uksw.edu.pl

Wydział Matematyczno-Przyrodniczy. Szkoła Nauk Ścisłych, UKSW

Tabele - gromadzenie danych

- ▶ Rekordy w tabeli mogą być gromadzone w sposób nieposortowany lub posortowany przy użyciu pewnego klucza
- ▶ Poprzez posortowanie rekordów przed wpisaniem ich na dysk, dostęp do bazy danych może zostać ulepszony
- ▶ Często zachodzi potrzeba posortowania i przeszukiwania rekordów przy użyciu różnych kluczy sortujących (np. po nazwisku klienta, po adresie itp.)

Indeks

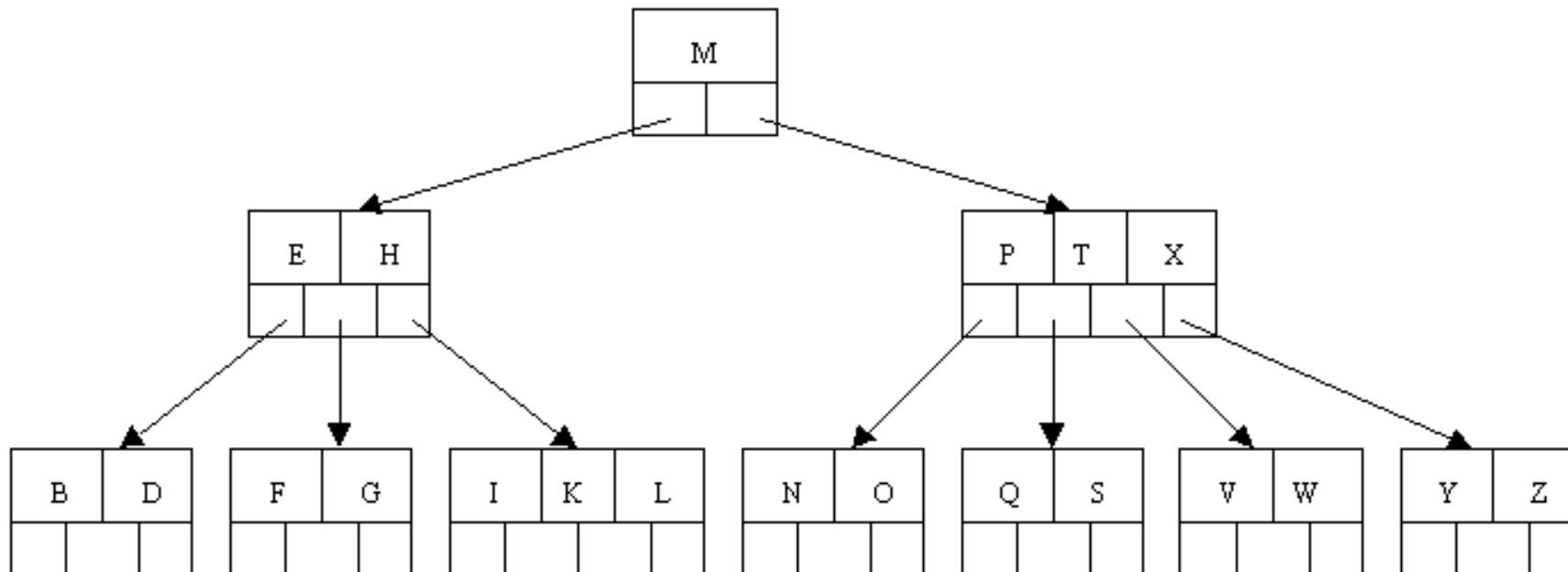
- ▶ Dodatkowa struktura używana, aby przyspieszyć wyszukiwanie informacji w tabeli poprzez dostarczenie drzewo podobnej struktury służącej do sortowania
- ▶ Zazwyczaj do przestawienia indeksu używa się B-drzewa
- ▶ Indeks zawsze odnosi się do danych z jednej tabeli
- ▶ Definicja indeksu ma zawsze formę
Column1 [ASC|DESC], Column2 [ASC|DESC],...
- ▶ Przykładowa definicja:
Kraj, Miasto, Adres

B-drzewa

- ▶ Definicja B-drzewa rzędu m ($m \geq 3$) jest następująca:
 - ▶ (1) Korzeń jest liściem, albo ma od 2 do m synów.
 - ▶ (2) Wszystkie liście są na tym samym poziomie.
 - ▶ (3) Każdy węzeł wewnętrzny oprócz korzenia ma od $\lceil m/2 \rceil$ do m synów. Węzeł mający l synów zawiera $l-1$ kluczy.
 - ▶ (4) Każdy liść zawiera od $\lceil m/2 \rceil - 1$ do $m-1$ kluczy.
- ▶ Warunki (3) i (4) gwarantują wykorzystanie przestrzeni dysku przynajmniej w ok. 50%
- ▶ Warunek (2) gwarantuje niewielką wysokość drzewa (w najgorszym razie ok. $\log_{m/2} n / (m/2)$, a w najlepszym ok. $\log_m n / m$ dla drzewa zawierającego n kluczy)

B-drzewo - przykład

- ▶ Jeśli mamy 1M rekordów i $b=100$ wpisów w bloku to dowolny rekord może być zlokalizowany poprzez co najwyżej 3 odczyty dysku
- ▶ Przykład 3-drzewo:



B-drzewo - dyskusja

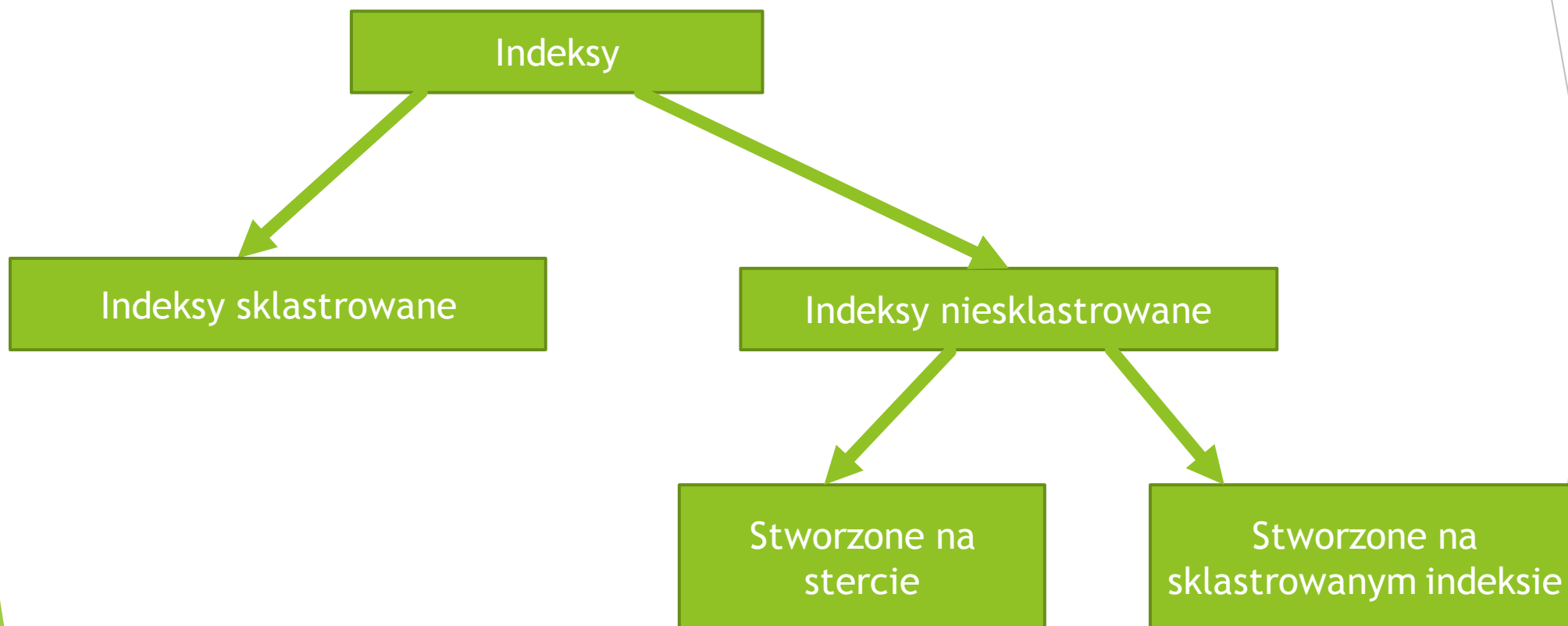
► Zalety:

- Utrzymuje klucze w uporządkowanej kolejności do przechodzenia sekwencyjnego
- Używa indeksu hierarchicznego, aby zminimalizować liczbę odczytów na dysku
- Używa częściowo pełnych bloków, aby przyspieszyć wstawianie i usuwanie
- Utrzymuje indeks zrównoważony algorytmem rekursywnym

► Wady:

- Nie można zmienić maksymalnej długości klucza bez całkowitego przebudowania bazy danych. Doprowadziło to w wielu SZBD obciążenia pełnych nazw ludzi (fullname) do 70 znaków.
- Wstawianie (po przekroczeniu pojemności bloku) i usuwanie (jeśli elementów w bloku jest za mało) może być bardzo kosztowne oraz wymagać przebudowy drzewa

Kategorie indeksów w MS SQL Server



Indeksy sklastrowane

- ▶ Zdefiniowanie indeksu sklastrowanego w tabeli oznacza zmuszenie SZBD, aby przechowywał fizycznie rekordy tabeli w specjalnym porządku
- ▶ Można stworzyć co najwyżej jeden indeks sklastrowany dla każdej tabeli
- ▶ Zwykle rekordy są przechowywane w B-drzewie, a liście drzewa zawierają strony danych

Indeks sklastrowany - przykład

- ▶ Przykład tabela SzczegółyZamówień z sklasteryzowanym indeksem na IdZamówienia

IdZamówienia	...	...	..
1890			
1891			
1892			
1893			

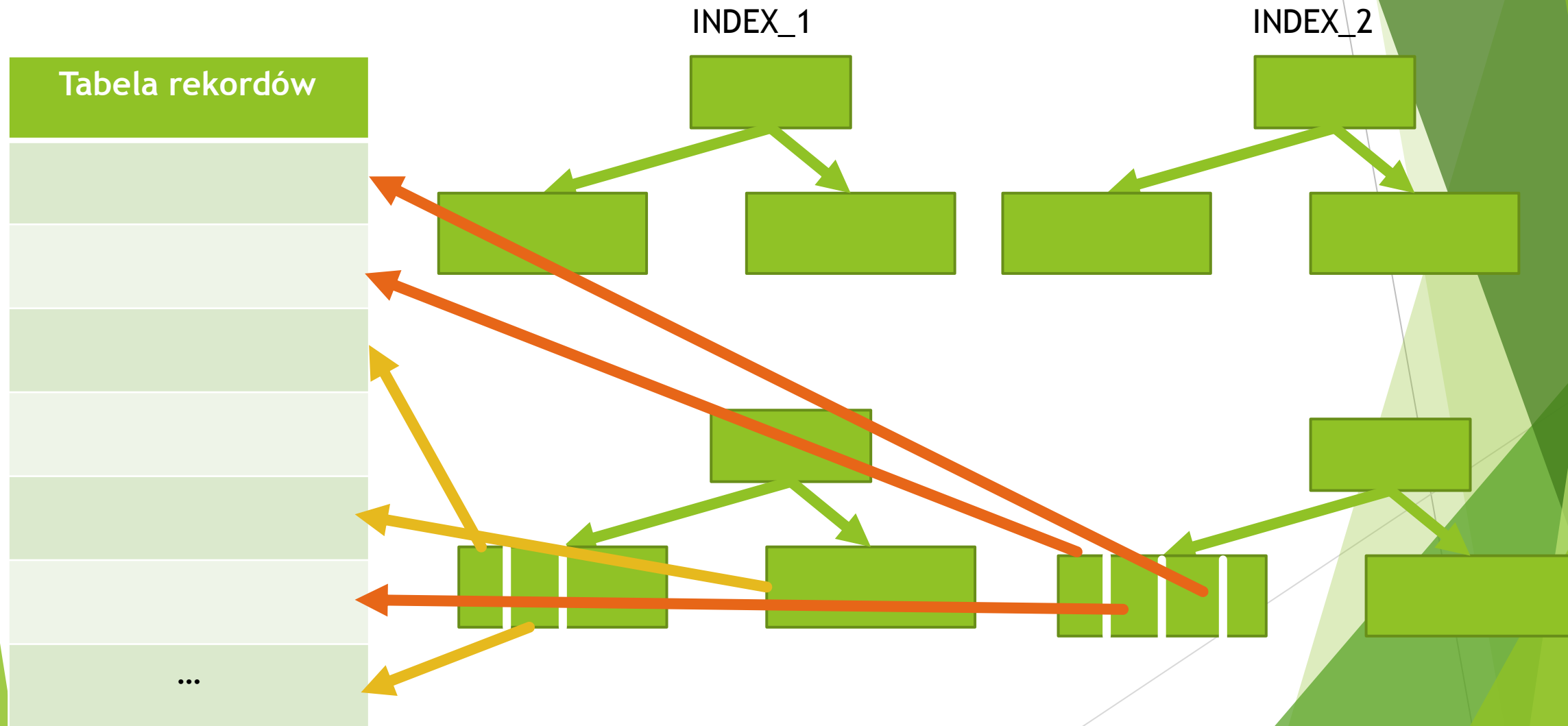
Sterta

- ▶ Jeśli żaden indeks sklastrowany nie został zdefiniowany w tabeli, rekordy są posortowane w formie sterty
- ▶ Nie istnieje żaden porządek występowania rekordów w tabeli
- ▶ Takie rekordy mogą być wstawione szybciej, jako że nie ma potrzeby przestrzegania żadnego porządku

Niesklastrowane indeksy

- ▶ W każdej tabeli może wystąpić wiele niesklastrowanych indeksów
- ▶ Przykład:
 - INDEX_1: Country, City, Address
 - INDEX_2: CustomerName
 - INDEX_3: CustomerID

Niesklastrowane indeksy



Niesklastrowane indeksy

- ▶ Drzewo indeksów zawiera jedynie wartość klucza indeksu, np. wartości kolumn zawierających klucz
- ▶ W ten sposób drzewa zawiera wskaźniki rekordów tabeli (a nie właściwe rekordy)

Zastosowania indeksów

- ▶ Zwiększa wydajność w przypadku wyszukiwania danych spełniających jakiś warunek (warunki WHERE) lub ich sortowania (ORDER BY)
- ▶ Przykład:
 - Indeks na Identyfikatorze zamówienia może zostać użyty do znalezienia wszystkich linii zamówień zawierających zamówienie przy użyciu drzewa indeksującego
 - Indeks na kraju, mieście, adresie może być użyty do sortowania klientów po kraju, mieście czy adresie

Negatywny wpływ indeksów

- ▶ Jeśli indeks jest stworzony, to SZBD staje się odpowiedzialna za aktualizowanie drzewa indeksującego, np. zgodnie z treścią tabeli
- ▶ Indeksy mogą wydłużyć czas wykonywania poleceń CRUD
- ▶ Polecenia CRUD może wymagać przebudowania wielu drzew indeksujących aby zaktualizować indeksy

Unikalne indeksy

- ▶ Mogą być stworzone przez SZBD aby zachować unikalność klucza indeksu (często podstawowy klucz tabeli)
- ▶ Przykład: unikalny indeks jest tworzony w kolumnie Identyfikator klienta w tabeli klientów
- ▶ Dlaczego?
 - SZBD podczas aktualizowania indeksu może z łatwością stwierdzić czy istnieje inny rekord o takiej samej wartości klucza indeksu
 - bez używania indeksu SZBD musiałaby przeszukiwać całą tabelę żeby znaleźć wartość konfliktową

Indeksy - zalecenia

- ▶ Prawdziwą zaletą indeksu jest zmniejszona liczba stron na dysku, które muszą zostać przeczytane
- ▶ Zamiast skanowania zawartości całej tabeli i czytania nawet megabajtów danych SZBD może przeczytać znacząco mniejszą liczbę bajtów definiujących strukturę indeksu
- ▶ Rozważ zdefiniowanie sklastrowanych indeksów w tabeli, gdy często korzystasz z grupy powiązanych rekordów, np. wiele linii zamówienia. Wtedy wszystkie wymagane rekordy będą przechowywane w tej samej części dysku.

Indeksy - kiedy ich używać?

- ▶ W każdym wypadku aby:
 - zachować unikalność podstawowego klucza
 - przyspieszyć sprawdzanie integralności referencji, np. indeksy powinny być stworzone na obcych kluczach
- ▶ Gdy oczekujemy dużej liczby rekordów w tabeli (>1000) i:
 - kolumna(/y) jest (są) często używana (/e) do operacji sortowania/ wyszukiwania
 - rozmiar klucza indeksu jest mały (nie tworzy się duże drzewo indeksowania)
 - istnieje dobra selektywność kolumn(y) indeksu

Integralność referencji a indeksy - przykład

Sale

Budynek	Piętro	Sala
WOY/21	0	033
WOY/21	1	134
...		...



klucz główny (PK - primary key)

Wyposażenie

Budynek	Zasób	Sala
WOY/21	Krzeseł	033
WOY/21	Stolik	033



klucz obcy (FK - foreign key)



- Indeks powinien zostać utworzony w kolumnach klucza podstawowego
- Indeks powinien być utworzony na kolumnach klucza obcego, w przeciwnym razie za każdym razem, gdy rekord w tabeli pokoi zostanie usunięty (lub zmodyfikowana wartość klucza), wszystkie rekordy w zasobach muszą być skanowane. Podobnie każdy JOIN będzie potrzebował skanowania tabeli podrzędnej.

Indeksy - kiedy unikać?

- ▶ Indeksy wydłużają czas przetwarzania zapytań CRUD, dlatego należy unikać stosowania niepotrzebnych indeksów
- ▶ Przykład:
 - Kolumn study_type przybiera jedynie dwie wartości: „Bachelor of Science” i „Master of Science” (bardzo niska selektywność), a więc:
 - ▶ średnio może to pomóc w przefiltrowaniu jedynie połowy rekordów,
 - ▶ wciąż zarówno drzewo indeksujące jako i połowa zawartości indeksu musi być wyszukiwana, co może wydłużyć czas przetwarzania.

Pytania?