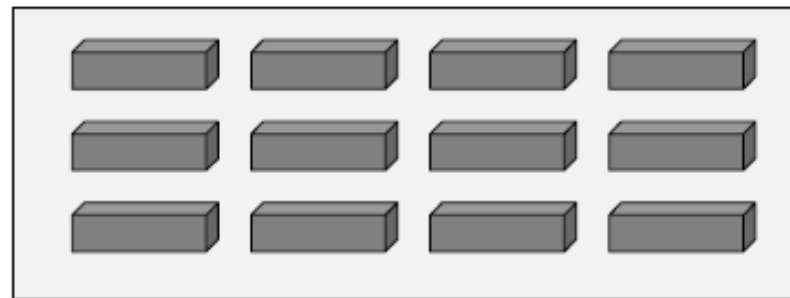


Struktury przechowywania danych

- » Dane do przetwarzania – pamięć główna
- » Trwałe dane w BD są przechowywane w pamięci zewnętrznej:
 - ze względu na rozmiar danych
 - odporność pamięci zewnętrznej na awarie
 - niski koszt przechowywania
- » Buforowanie bloków dyskowych
 - dane odczytywane z plików do bloków (buforów) BD
 - udostępniane użytkownikom BD (odczyt/modyfikacje)
 - zapis do plików na dysku buforów BD

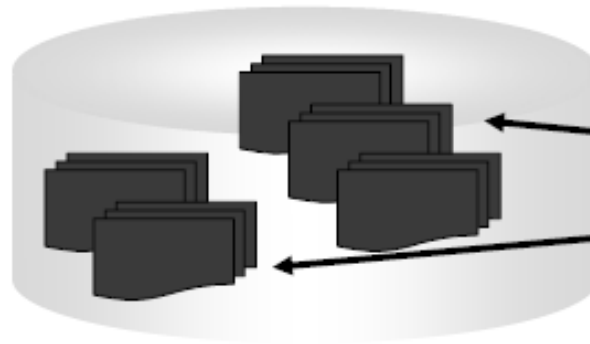
- » Media fizyczne tworzą hierarchię pamięci składającą się z:
- pamięci operacyjnej o organizacji blokowej
 - pamięci zewnętrznej o organizacji plikowej

pamięć
operacyjna



bloki
(bufory)

dysk



pliki
(strony)

Organizacja pliku

- » *Blok* może zawierać wiele rekordów
 - założenie:
 - rekord nie jest dłuższy od bloku
 - każdy rekord zawiera się w całości w bloku
- » Najprostsze podejście:
 - założenie, że wielkość rekordu jest stała
 - każdy plik ma rekordy tylko jednego typu
 - różne pliki do różnych relacji

Rekordy stałej długości - przykład

- » Plik zawiera rekordy *instructor*, każdy rekord:

```
type instructor = record
```

```
    ID varchar(5);
```

```
    name varchar(20);
```

```
    dept_name varchar(20);
```

```
    salary numeric(8,2);
```

```
end
```

- » każdy rekord zajmuje 53 bajty (1 znak= 1 bajt, numeric = 8 bajtów)

Rekordy stałej długości

- Podejście:

- Przechowuj rekord i zaczynając od bajtu $n * (i - 1)$, gdzie n jest rozmiarem każdego rekordu.
- Dostęp do rekordu prosty, ale rekordy mogą zajmować wiele bloków

- podejście: pozostawić puste miejsce w bloku

instructor

- Usuwanie rekordu i (trudne):
podejścia:

- przesun rekordy $i + 1, \dots, n$ do $i, \dots, n - 1$
- przesun rekord n do i
- nie przesuwaj rekordów, ale zlinkuj wszystkie wolne rekordy w *free list*

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Usunięcie rekordu 3 i kompaktowanie

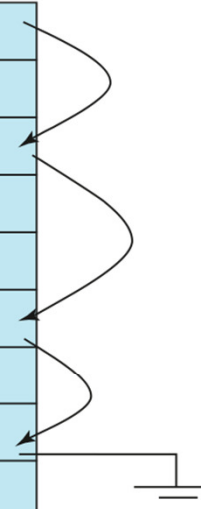
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Usunięcie rekordu 3 i przesunięcie ostatniego

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 11	98345	Kim	Elec. Eng.	80000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000

- Przechowaj adres pierwszego skasowanego rekordu w nagłówku pliku.
- Użyj pierwszy rekord do przechowania adresu drugiego skasowanego rekordu itd.
- Można traktować przechowywane adresy jako **wskaźniki** gdyż wskazują na lokalizację rekordu (free list).
- Wstawianie nowego rekordu: wykorzystanie pierwszego z *free list* oraz aktualizacja adresu w nagłówku

header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1				
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000



Reprezentacja rekordu (atrybuty stałej i zmiennej długości)

- Przyczyny występowania rekordów o zmiennej długości:
 - przechowywanie różnych typów rekordów w pliku
 - typy rekordów pozwalające na zmienną długość dla jednego lub większej liczby pól (**`varchar`**)
- Różne techniki do implementacji rekordów zmiennej długości.
- Problemy do rozwiązania:
 - jak reprezentować pojedynczy rekord aby można było prosto wyekstrahować pojedyncze atrybuty
 - jak przechowywać rekordy zmiennej długości w bloku, tak by rekordy z bloku było łatwo wyekstrahować



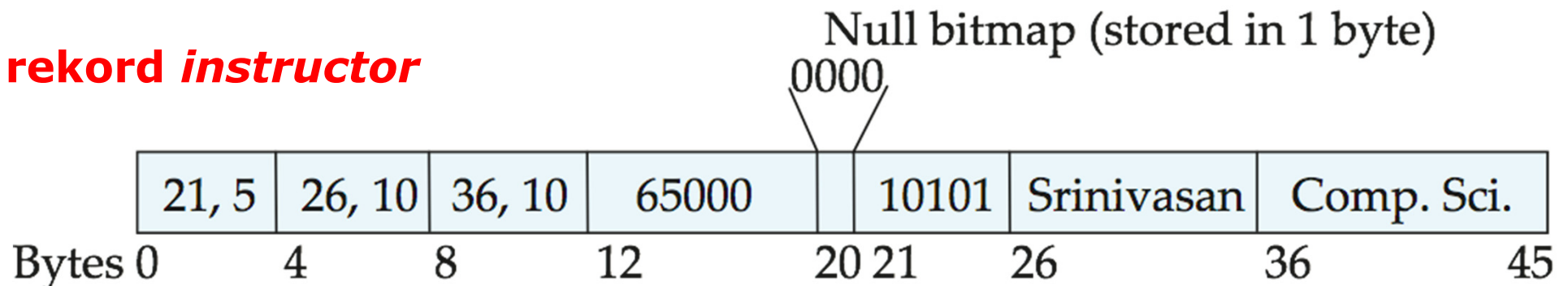
AGH

Reprezentacja rekordu (atrybuty stałej i zmiennej długości) (1)

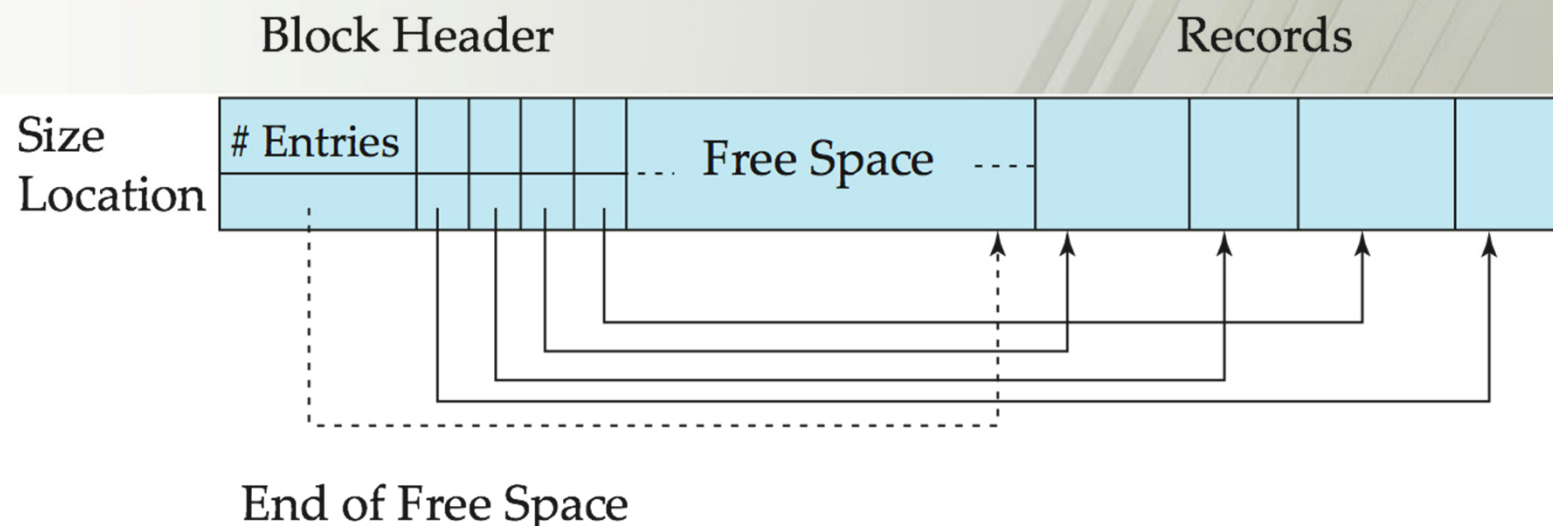
```
type instructor = record (ID varchar(5);  
                           name varchar(20);  
                           dept_name varchar(20);  
                           salary numeric(8,2);end
```

- Złożona z 2 części:
 - część początkowa z atrybutami stałej długości
 - dane dla atrybutów zmiennej długości
- Atrybuty stałej długości (liczby, daty, stringi) – alokowane tyle bajtów ile wymaga długość
- Atrybuty zmiennej długości (varchar) reprezentowane w początkowej części rekordu przez parę: (offset, długość), z aktualnymi danymi przechowywanymi po wszystkich atrybutach stałej długości
- Wartości *null* reprezentowane przez *null bitmapę* (czasem na początku)

rekord *instructor*



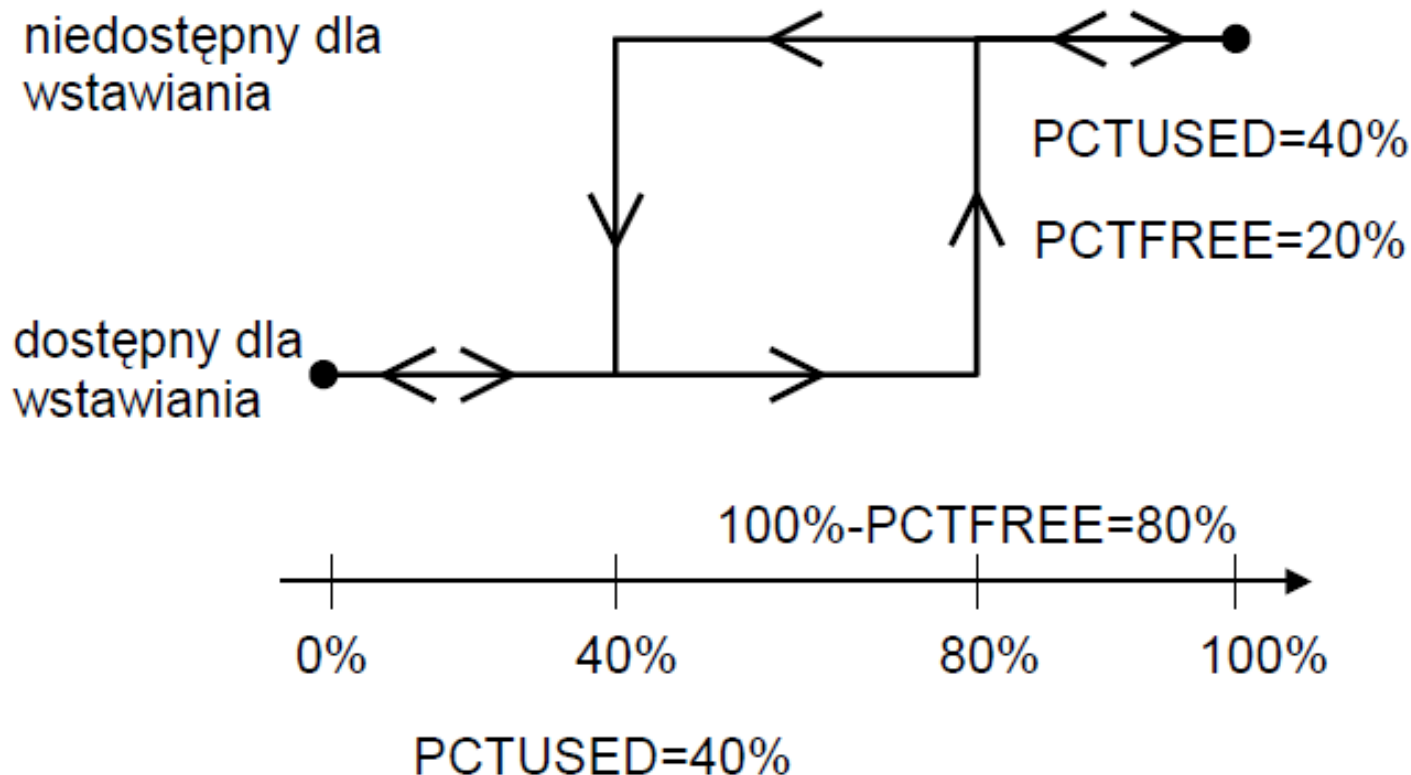
Przechowywanie rekordów zmiennej długości w bloku: Struktura Slotowa Strony (Slotted Page Structure)



- Nagłówek **slotted page** zawiera:
 - liczba rekordów
 - koniec wolnej przestrzeni w bloku
 - tablica z lokalizacjami i rozmiarem każdego rekordu
- Wstawianie rekordów
- Usuwanie rekordów – inne przesuwane wraz z aktualizacją nagłówka (małe koszty)
- Brak konieczności przechowywania wskaźników bezpośrednio do rekordów

Zarządzanie rozmiarem bloku danych

- » Utrzymywanie wolnej pamięci w bloku dla potencjalnej modyfikacji
 - PCTUSED – kiedy do bloku można wstawiać rekordy (%rozmiaru bloku)
 - PCTFREE – ile % rozmiaru bloku pozostanie wolne



Jak organizować zbiór rekordów w pliku

- **Nieuporządkowane** – rekordy przechowywane w dowolnej kolejności
- **Uporządkowane (sekwencyjne)** – rekordy wstawiane w porządku sekwencyjnym, w oparciu o wartość klucza wyszukiwania
- **Haszowe** – funkcja haszująca wyliczana na określonym atrybucie w każdym rekordzie; wynik wskazuje w którym bloku w pliku należy umieścić rekord
- **Multitable clustering** (klastrowanie wielotablicowe)
rekordy wielu różnych relacji mogą być przechowywane w tym samym pliku
 - Motywacja: przechowywanie powiązanych rekordów w tym samym bloku minimalizuje I/O



AGH

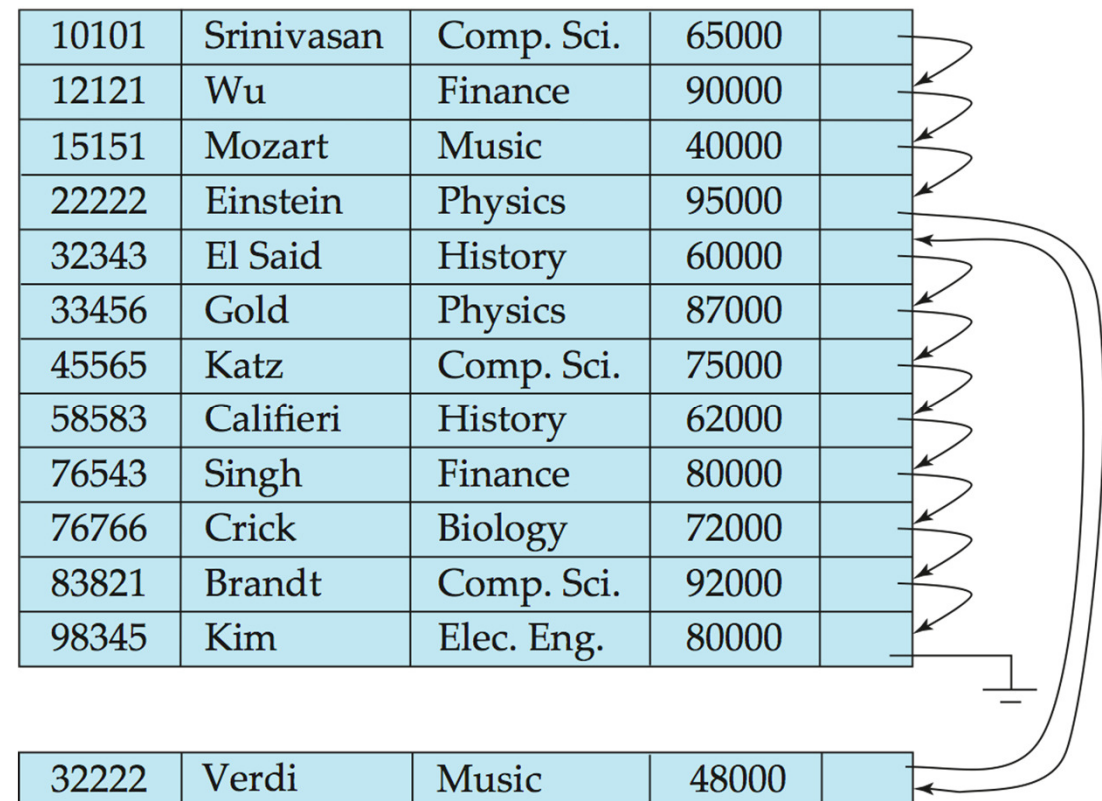
Organizacja pliku sekwencyjnego

- Przydatne dla aplikacji wymagających sekwencyjnego przetwarzania całego pliku
- Rekordy w pliku fizycznie uporządkowane wg **klucza wyszukiwania (search-key)**
- Pointer wskazuje na następny rekord wg klucza wyszukiwania

10101	Srinivasan	Comp. Sci.	65000	
12121	Wu	Finance	90000	
15151	Mozart	Music	40000	
22222	Einstein	Physics	95000	
32343	El Said	History	60000	
33456	Gold	Physics	87000	
45565	Katz	Comp. Sci.	75000	
58583	Califieri	History	62000	
76543	Singh	Finance	80000	
76766	Crick	Biology	72000	
83821	Brandt	Comp. Sci.	92000	
98345	Kim	Elec. Eng.	80000	

Organizacja pliku sekwencyjnego c.d.

- Usuwanie – wykorzystanie łańcucha wskaźników
- Wstawianie – zlokalizuj pozycję, gdzie rekord powinien być wstawiony
 - jeżeli jest wolne miejsce, wstaw tam
 - jeżeli brak wolnego miejsca, wstaw rekord do **overflow block**
 - w każdym przypadku, łańcuch wskaźników musi być uaktualniony
- Potrzeba okresowej reorganizacji pliku w celu odtworzenia porządku sekwencyjnego
- Trudne zarządzanie

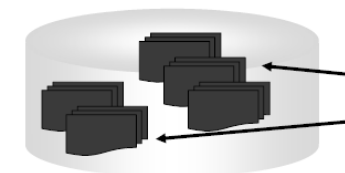


Dostęp do pamięci

pamięć
operacyjna



dysk



- Plik bazy danych podzielony na stałej długości jednostki pamięci zwane **blokami**
 - jednostki zarówno alokacji pamięci jak i transferu danych.
- Cel SZBD: minimalizacja liczby transferów bloków między dyskiem a pamięcią
 - podejście: przechowywanie jak największej liczby bloków w pamięci głównej.
- **Bufor (pula buforów)** – porcja pamięci głównej przeznaczona do przechowywania kopii bloków dyskowych.
- **Zarządca bufora (buffer manager)** – podsystem odpowiedzialny za alokację przestrzeni buforów w pamięci głównej.

Zarządca bufora

- Programy wysyłają żądanie do zarządcy bufora gdy potrzebują blok z dysku.
 1. Jeżeli blok jest już w buforze, zarządca zwraca adres bloku w pamięci głównej
 2. Jeżeli nie, to zarządca:
 1. alokuje przestrzeń w puli buforów dla bloku
 1. Wymiana (wyrzucenie) jakiegoś innego bloku, w razie potrzeby, aby zrobić miejsce na nowy blok
 2. Zastąpiony blok zapisywany na dysk tylko wtedy gdy został zmodyfikowany od ostatniego czasu gdy został zapisany/pobrany na/z dysk
 2. czyta blok z dysku do bufora i zwraca adres bloku w pamięci głównej żądającemu.

- Gdy brakuje miejsca w buforze – najczęściej usuwany blok najdawniej używany (**LRU strategy**)
- **Przypięty blok (pinned block)** – blok pamięci, którego nie można zapisać z powrotem na dysk.
- Zarządca buforów wspiera również **wymuszony zapis (forced output)** bloków, nawet jeżeli nie brakuje miejsca w buforze

Buforowanie bloków dyskowych

