

データベースシステム

コストの考え方とデータの格納方式

内容

1. 高速な検索を実現するために必要な技術
2. 検索時間を考えるための重要な用語
「IOコスト」
3. もっとも簡単なデータ格納方式
「ヒープファイル」
4. 問合せの種類別, IOコストの計算の仕方
～ ヒープファイルを使って ～
5. IOコストの比較
～ ハッシュファイルと比較してみよう ～



問合せの例

- ▶ g0720501が履修する授業名を求めるには？

```
SELECT 授業.授業名
FROM 授業,履修
WHERE 授業.授業番号 = 履修.授業番号
      and 履修.学籍番号 = 'g0720501'
```

授業

授業番号	授業名
PR001	プログラミング実習
IT002	情報理論
AI003	人工知能論

履修

授業番号	学籍番号	成績
PR001	g0720501	A
PR001	g0720504	B
IT002	g0720502	A
IT002	g0720507	A
AI003	g0720503	B

検索を速くするための技術を学ぶ

- ▶ g0720501が履修する授業名を求めるには？

```
SELECT 授業.授業名
FROM 授業, 履修
WHERE 授業.授業番号 = 履修.授業番号
      and 履修.学籍番号 = 'g0720501'
```

授業

授業番号	授業名
PR001	プログラミング実習
IT002	情報理論
AI003	人工知能論

履修

授業番号	学籍番号	成績
PR001	g0720501	A
PR001	g0720504	B
IT002	g0720502	A
IT002	g0720507	A
AI003	g0720503	B

検索の重要ポイント（その1）

▶ 条件による選択

```
SELECT 授業.授業名  
FROM 授業, 履修  
WHERE 授業.授業番号 = 履修.授業番号  
and 履修.学籍番号 = 'g0720501'
```

選択の条件

授業

授業番号	授業名
PR001	プログラミング実習
IT002	情報理論

履修

授業番号	学籍番号	成績
PR001	g0720501	A
PR001	g0720504	B
		A
		A
		B

履修テーブルにタプルが1000万あったとき
どうしたら一瞬で検索することができるだろうか？

検索の重要ポイント（その1）

- ▶ 条件による選択を高速に行うには
 1. 頻繁に発行される問合せ条件を調べて置くこと
 - ▶ アプリケーションで発行される問合せの種類は大体決まっている
 2. 問合せ条件に合った格納方式を選ぶ
 3. 問合せ条件に合った索引を付ける



検索の重要ポイント（その2）

▶ 結合演算

```
SELECT 授業.授業名  
FROM 授業, 履修  
WHERE 授業.授業番号 = 履修.授業番号
```

授業

授業番号	授業名
PR001	プログラミング実習
IT002	情報理論
AI003	人工知能論

履修

授業番号	学籍番号	成績
PR001	g0720501	A
PR001	g0720504	B
IT002	g0720502	A
IT002	g0720507	A

履修テーブルが1000万、授業テーブルが1000あっても検索を一瞬で終わらせることができる？

検索の重要なポイント（その2）

▶ 結合演算

1. 結合アルゴリズムを知る
2. 適切な結合アルゴリズムが実行されるように、索引などをあらかじめ用意しておく



高速な検索を実現するために

データの格納方式と選択処理

- どんな選択条件のときにどの格納方式がよいか

結合処理のアルゴリズム

問合せのプラン組立と最適化



データ格納方式と選択処理

演習のための準備

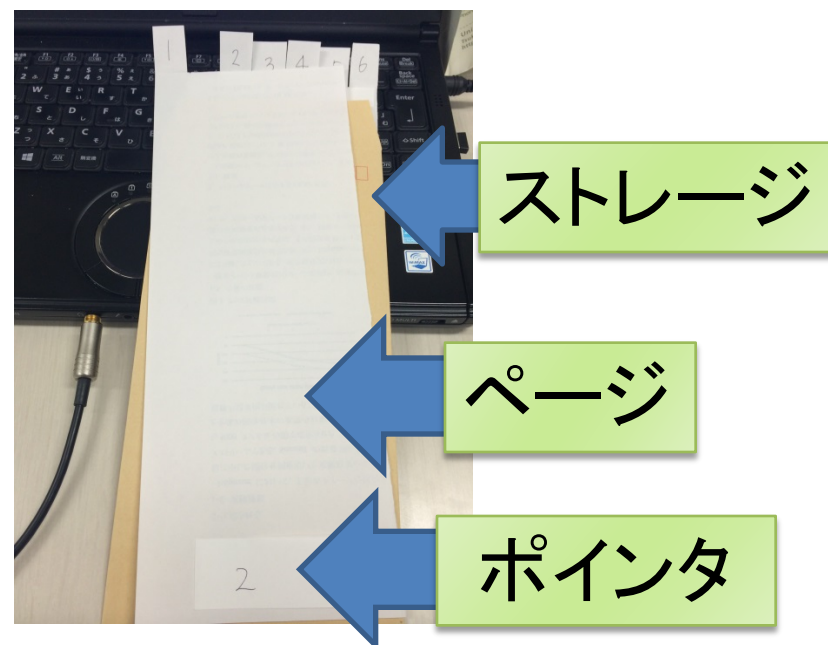
- ▶ 隣 & 後ろにいる人で4人組を作しましょう
- ▶ 4人組を2人ずつに分けましょう
 - ▶ 1組目：データベースを作る
 - ▶ 2組目：データを用意する



演習のための準備（その1）

▶ 簡易データベースシステムを作ります

- ▶ A4用紙3枚をそれぞれ縦半分に切る
- ▶ 上に切れ目を入れて「タブ」を作ってください
 - ▶ 1~5まで区別がつくように
- ▶ ~~小さい付箋を下に張ってください~~
- ▶ ~~付箋に以下の情報を書く~~
 - ▶ ~~「次のページ番号」~~
- ▶ 一旦封筒に入れます
 - ▶ タブ部分を引っ張って引き出すことができればOK



準備2: データベースにデータを挿入する

- ▶ 付箋に「名前」と「1～100の数字」を書きます
 - ▶ 名前はアルファベットで書いてください
 - ▶ 数字はランダムに書きましょう
 - ▶ 付箋を「レコード」と呼ぶことにします
- ▶ レコード10枚分書きましょう
- ▶ 1ページに3枚まで貼り付けてデータベースにしまってください
 - ▶ ページ1から詰めて貼り付けてください
 - ▶ あくまで数字はランダムな並びになるように



演習 1：検索速度を測りましょう

▶ 問合せ

- ▶ 数字が**35以下**のレコードをみつけましょう
(ページからはがして机に並べましょう)

▶ 検索のためのルール

- ▶ 検索作業をするのは「**データベースを作った**」組
 - ▶ データを用意した組は**時間を測りましょう**
- ▶ 封筒の外に出せるページは**1ページのみ**
 - ▶ 次のページを引き出すときは、外に出ているページをしまってから
- ~~▶ 次に引き出すことのできるページは、今出ているページ上の**ポインタ(付箋)**で示す**ページのみ**~~
 - ~~▶ つまり今のところは1→2→3→4→5→6~~



演習 1：振り返り

- ▶ 問合せに該当する件数
- ▶ かかった時間
- ▶ 何処で時間がかかりましたか？
- ▶ 他に気が付いたことがあれば書いてみましょう



格納するときの基本事項（レコード）

▶ その1

テーブルのタプル単位で保存する。
その時のタプルを「レコード」と呼ぶ

社員番号	社員名	趣味
0650	鈴木一郎	野球
0650	鈴木一郎	盆栽
0650	鈴木一郎	コイン収集
1508	浜崎アユ	作詞

0650	鈴木一郎	野球
------	------	----

0650	鈴木一郎	盆栽
------	------	----

0650	鈴木一郎	コイン収集
------	------	-------

1508	浜崎アユ	作詞
------	------	----

格納するときの基本事項（ページ）

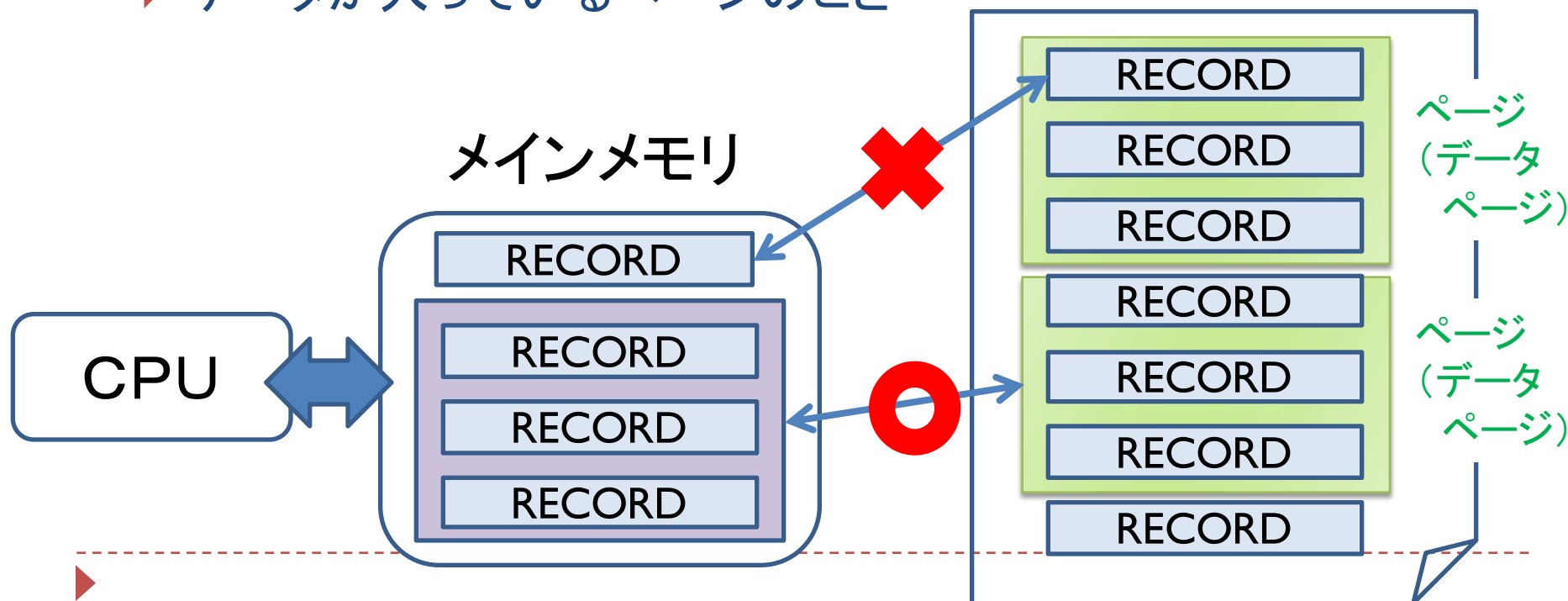
▶ その2

ディスク(ファイル)からの読み込みや書き込みはレコード単位では行わず「ページ」単位で行う

▶ データページ

▶ データが入っているページのこと

ファイル(in ディスク)



適切なページの大きさ

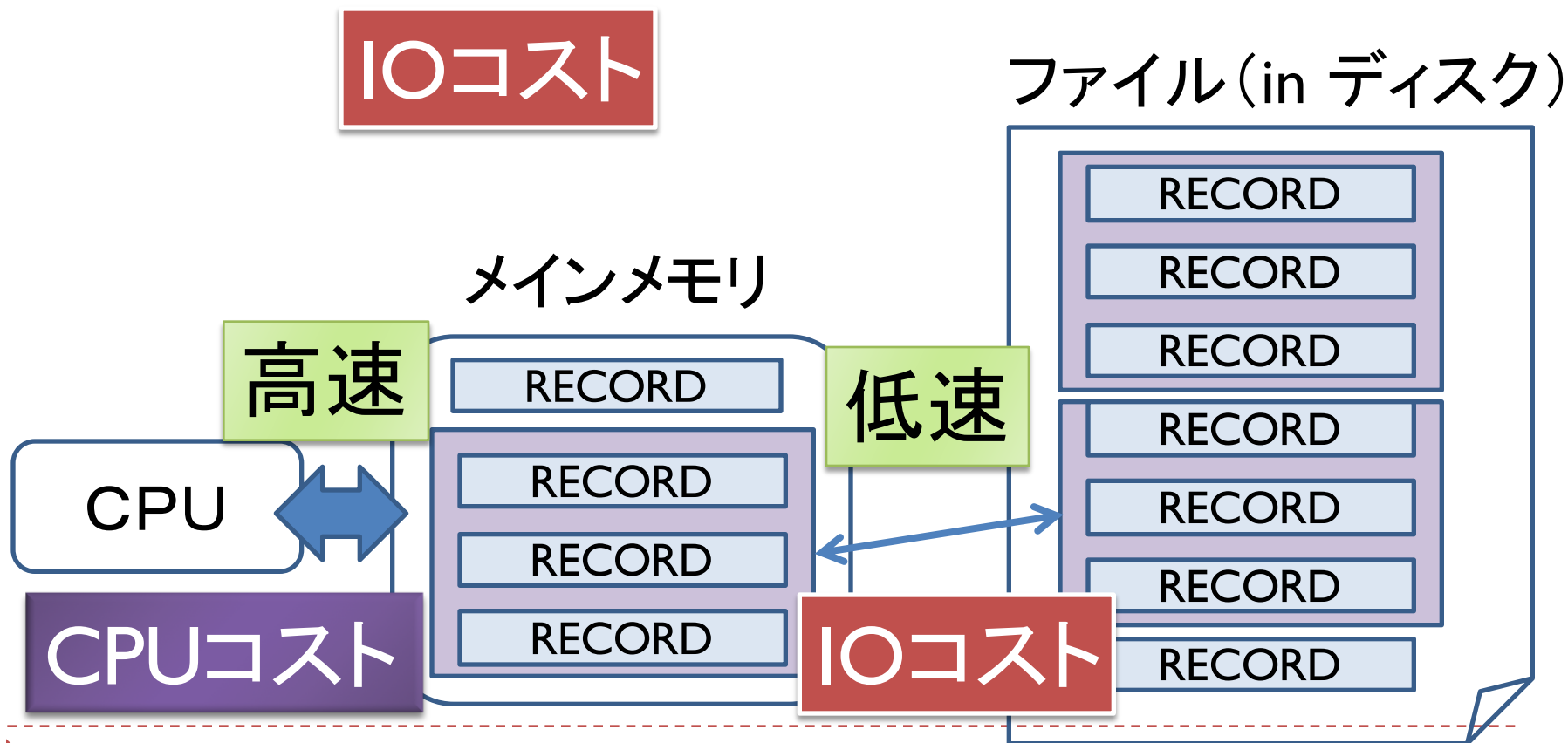
- ▶ 殆どのデータベースシステムで、ページサイズは8MBに設定されている
- ▶ もちろん設定で変更可能
- ▶ ページが大きすぎると...
 - ▶ たった1つのタプルを読み込みただけなのに、たくさんのデータをファイルから読み込まないといけない
- ▶ ページが小さすぎると...
 - ▶ 全てのタプルを読み込むために何度もディスクにアクセスしなければならない



格納するときの基本事項 (IOコスト)

▶ その3

アクセスの効率を考えるときはディスク
(ファイル)に何回読み書きをしたかを基準にする

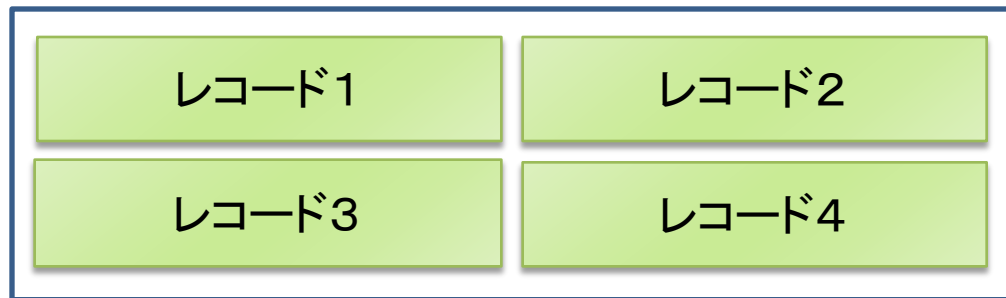


もっとも単純な格納方式
ヒープファイル

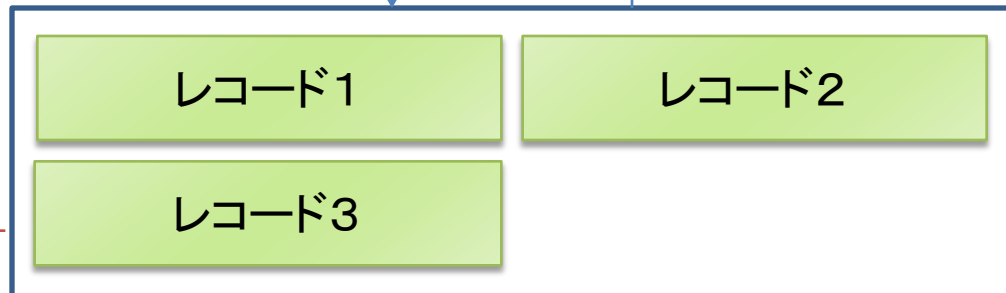
ヒープファイル

- ▶ もっとも基本的な格納の仕方
- ▶ ページにレコードを順次格納したファイル
- ▶ ページがレコードでいっぱいになったら新しいページを作る

ページ1



ページ2



問合せの種類別IOコストの計算の仕方 ～ ヒープファイルを使って～

データベースに対する問合せ（選択演算）

スキャン

- 全部のデータを読み込む

範囲問合せ

- $\text{age} > 10$ というような比較条件を用いたもの

完全一致問合せ

- 等号を使った問合せ ($\text{id} = \text{'g0520434'}$)
- 答えは1つとは限らない



データベースに対する問合せ（更新処理）

挿入

- 1つのレコードを挿入する

削除

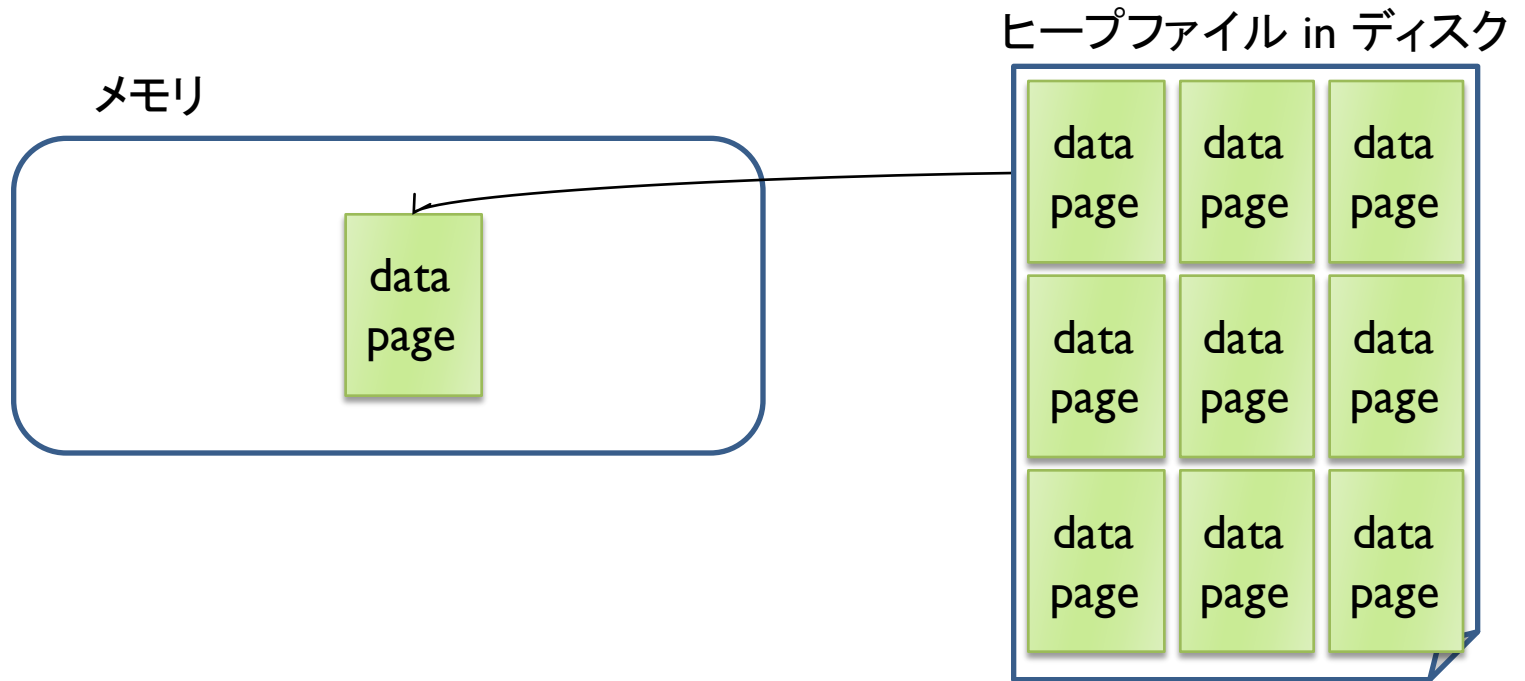
- 1つのレコードを削除する

更新

- 1つのレコードの属性「氏名」を変更する



ヒープファイルをつかったスキャン



- (1) 最初のデータページを取り出し、レコードを順に取り出す
- (2) 全部のレコードを取り出し終わったら、次のデータページのアドレスを調べる(ページの中に書かれている)
- (3) (2)で調べたアドレスから次のページを取り出す
- (4) 次のデータページのアドレスがNULLになるまで(2),(3)を繰り返す

演習 2

- ▶ 以下の選択演算はどのように行うか処理の手順を書いてみましょう。また、ヒープファイルにNページあった時コストはどのくらいかかるか求めましょう

スキャン

- 全部のデータを読み込む

範囲問合せ

- $\text{age} > 10$ というような比較条件を用いたもの

完全一致問合せ

- 等号を使った問合せ ($\text{id} = \text{'g0520434'}$)
-



演習 3

- ▶ 以下の演算はどのように行うか処理の手順を書いてみましょう。また、ヒープファイルにNページあった時コストはどのくらいかかるか求めましょう

挿入

- 1つのレコードを挿入する

削除

- 1つのレコードを削除する

更新

- 1つのレコードの属性「氏名」を変更する
-



ヒント：削除処理

1. 対象レコードが見つかるまで読み込む

▶ 最良の場合

- ▶ 1ページ目で発見される

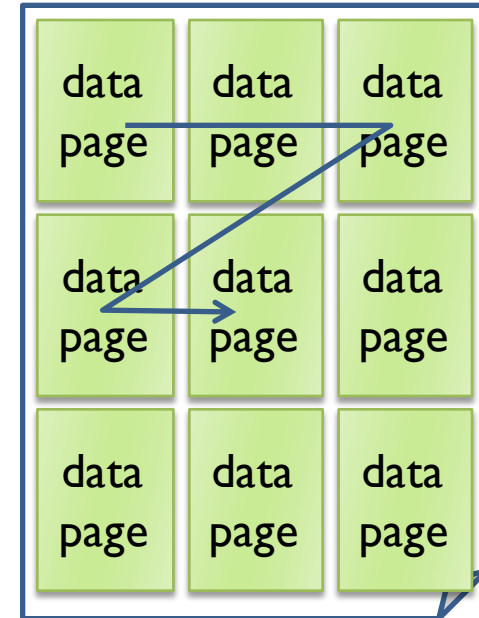
▶ 最悪の場合

- ▶ 最後まで見つからない

▶ IOコストは平均で考える

2. 対象レコードを書き換えてディスクに保存する

▶ 書込みコストは1



次の演習（予告）

- ▶ 検索を速くするための仕組みを考えてみましょう
 - ▶ ポインタの大きさはレコードの1/3
 - ▶ 1ページはポインタ11枚分の大きさがある
- ▶ ポインタに書けるもの
 - ▶ 数字, 数字, 数字
 - ▶ 文字, 数字, 数字



演習 4：検索速度を測りましょう

▶ 問合せ

- ▶ 文字がt(T)から始まるレコードをみつけましょう
(ページからはがして机に並べましょう)

▶ 検索のためのルール

- ▶ 検索作業をするのは「データベースを作った」組
 - ▶ データを用意した組は時間を測りましょう
- ▶ 封筒の外に出せるページは1ページのみ
 - ▶ 次のページを引き出すときは、外に出ているページをしまってから
- ~~▶ 次に引き出すことのできるページは、今出ているページ上のポインタ(付箋)で示すページのみ~~
 - ~~▶ つまり今のところは1→2→3→4→5→6~~



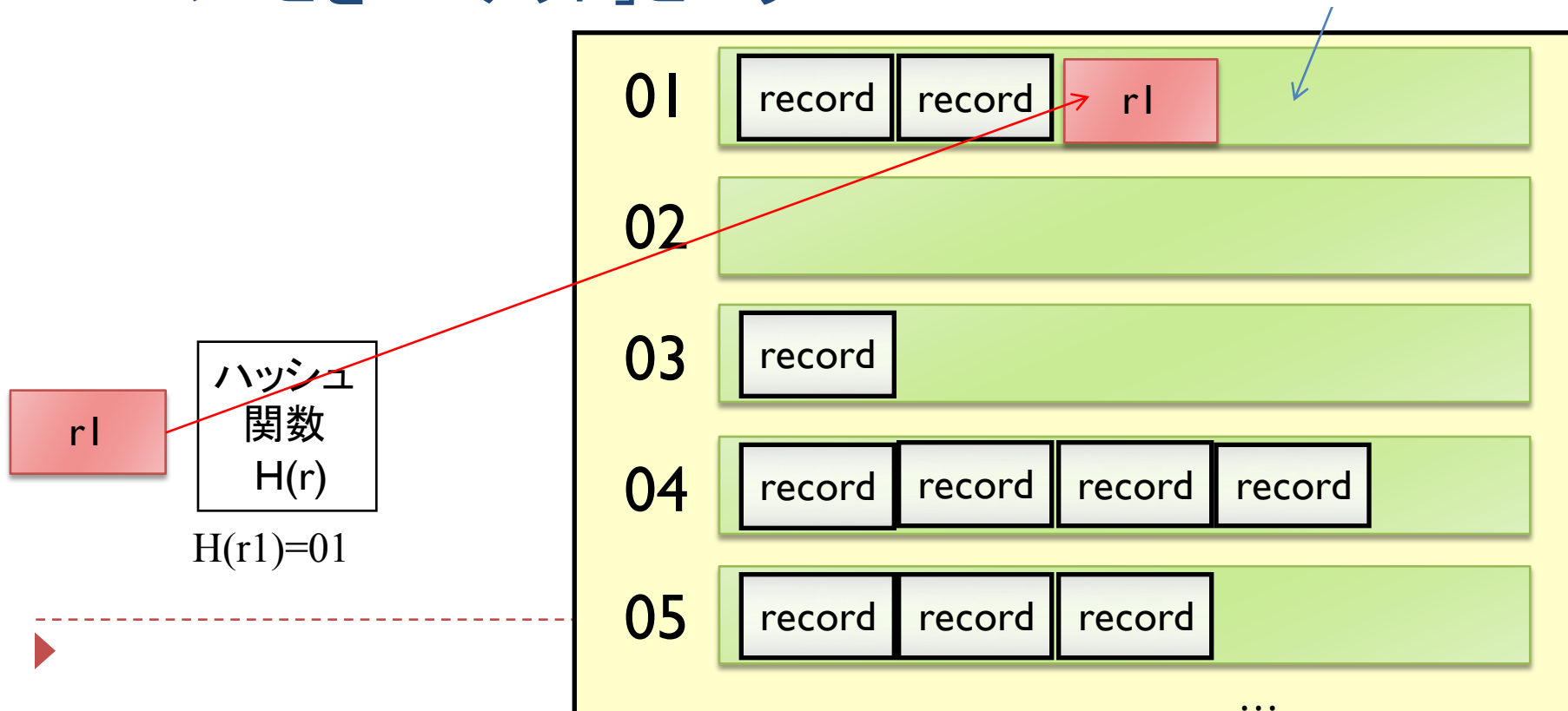
IOコストの比較 ~ ハッシュファイルとの比較 ~

ハッシュファイル

- ▶ レコードRのキー値を適当なハッシュ関数を用いてハッシングし、格納するバケットを決定する

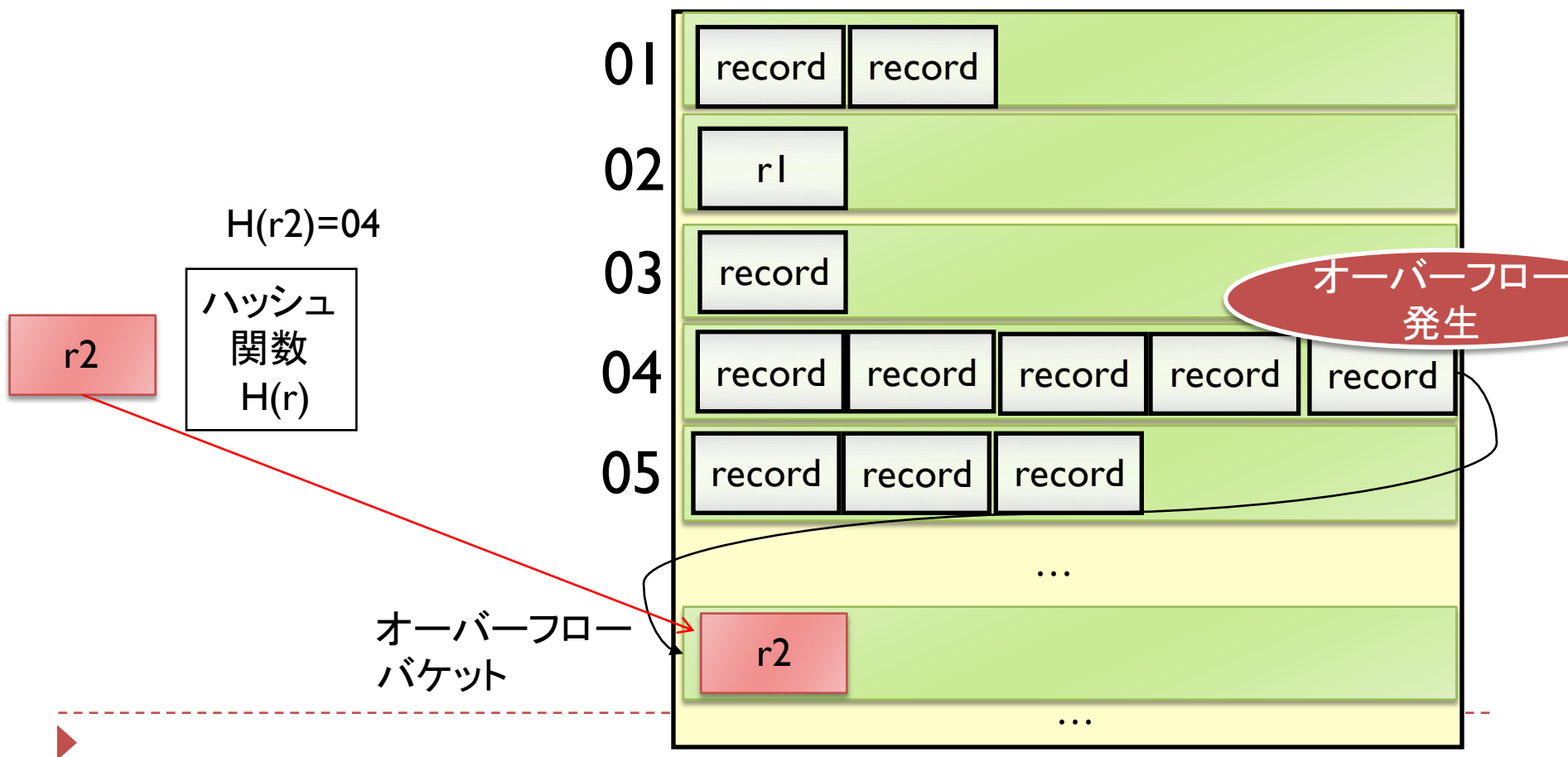
- ▶ ハッシュファイルの場合ページのことを「バケット」という

バケット(=データページ)



レコードがいっぱいになった場合

- ▶ オーバーフローバケットを用意して、そこにレコードを追加する



バケットの数とハッシュ関数

- ▶ バケットの数は2のべき乗で、レコード数によって適切なバケット数を設定すべき
 - ▶ バケット数が多すぎると
 - ▶ 殆どレコードが入っていないバケットばかりでディスクに無駄な領域を確保しなければならない
 - ▶ バケット数が少なすぎると
 - ▶ すぐにバケットがいっぱいになり、オーバーフローが頻発する



ハッシュファイルに対する問合せ（選択演算）

スキャン

- 全部のデータを読み込む

範囲問合せ

- $\text{age} > 10$ というような比較条件を用いたもの

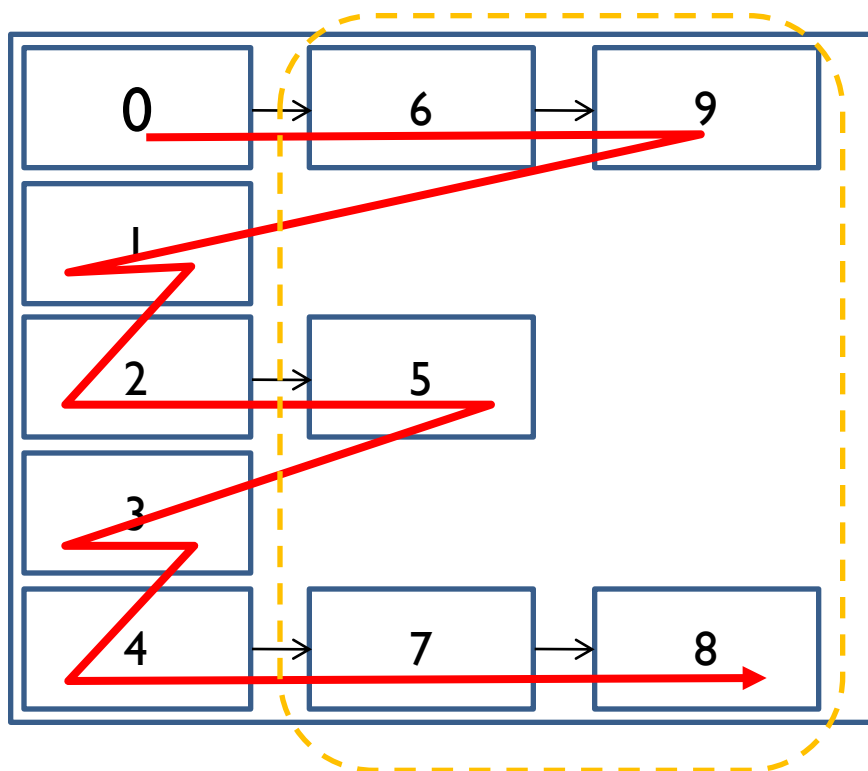
完全一致問合せ

- 等号を使った問合せ ($\text{id} = \text{'g0520434'}$)
- ハッシュ値はidをもとに求められるとする
- 答えは1つとは限らない



ハッシュファイルに対するスキャン

- ▶ ハッシュファイルに入っているバケットを上から順にメモリに呼び出してレコードを読み込む



オーバーフローバケット

ハッシュファイルのバケット数見積もり

▶ 問題

- ▶ ヒープファイルにレコードがぎっしり詰まってNバケット(←修正)になった時、それと同数のレコードをハッシュファイルに入れたら、ハッシュファイルのバケット数はどのくらいになるだろうか？
 - ▶ ヒント: 平均的に各バケットに5/6程度レコードが詰まっているとする

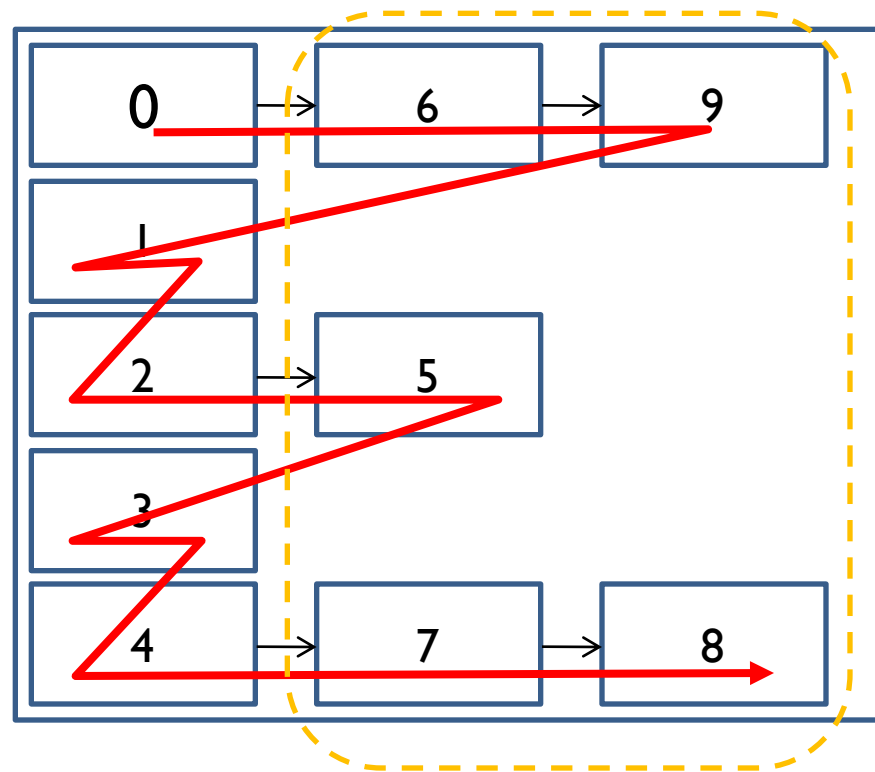
▶ 解答

- ▶ A : N個
- ▶ B : 1.2 N個
- ▶ C : 2N個



ハッシュファイルに対する範囲問合せ

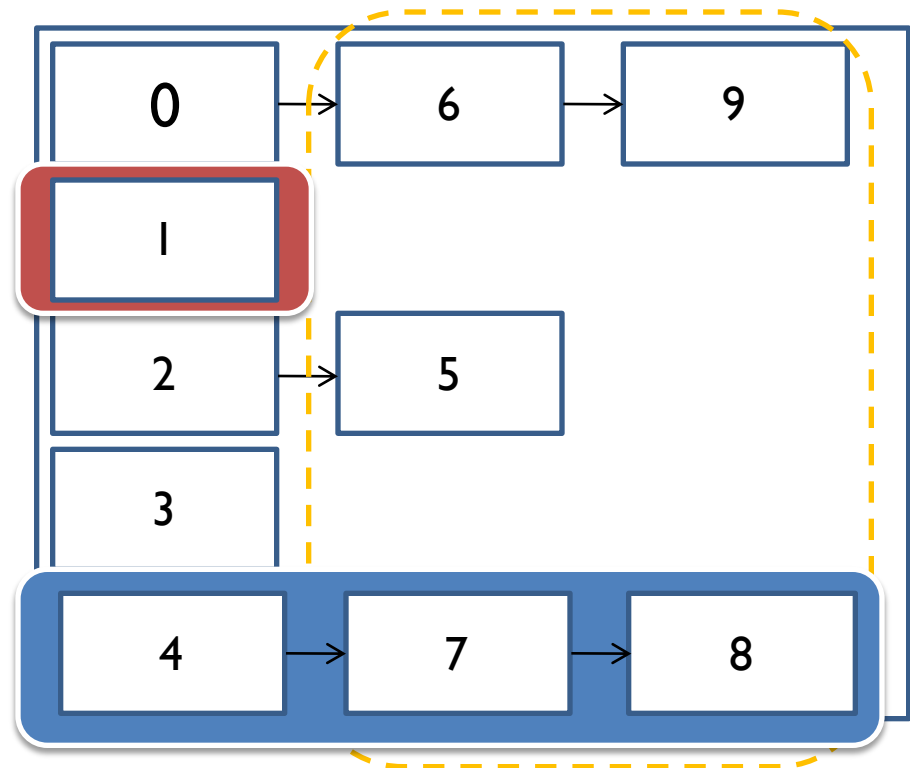
- ▶ $\text{age} > 10$ というような比較条件を用いたもの
- ▶ 全てのバケットを上から順にメモリに呼び出してレコードを読み込み条件に合うかどうか調べる



オーバーフローバケット

ハッシュファイルに対する完全一致問合せ

- ▶ 検索するid番号からハッシュ値を求める
- ▶ ハッシュ値に該当するバケットを順番に読みだす
- ▶ 最良の場合
 - ▶ 1つのバケットしかない
- ▶ 最悪の場合
 - ▶ オーバーフロー
バケットがたくさん
 - ▶ 平均はとりづらいので
オーバーフロー分を
 α としてコスト計算



演習 3

- ▶ 以下の選択演算はどのように行うか処理の手順を書いてみましょう。また、ハッシュファイルにNページあった時コストはどのくらいかかるか求めましょう

スキャン

- 全部のデータを読み込む

範囲問合せ

- $\text{age} > 10$ というような比較条件を用いたもの

完全一致問合せ

- 等号を使った問合せ ($\text{id} = \text{'g0520434'}$)
-

演習 4

- ▶ 以下の演算はどのように行うか処理の手順を書いてみましょう。また、ハッシュファイルにNページあった時コストはどのくらいかかるか求めましょう

挿入

- 1つのレコードを挿入する

削除

- 1つのレコードを削除する

更新

- 1つのレコードの属性「氏名」を変更する
-

演習 5

- ▶ ヒープファイルにNページあった場合の二つの格納形式のコストを比較し、どの処理にはどちらが向いているかを分析しよう

	ヒープファイル	ハッシュファイル
スキャン	N	$\frac{6}{5}N$
範囲問合せ	N	$\frac{6}{5}N$
完全一致	N	$1 + \alpha$
挿入	2	$1 + \alpha + 1$
削除	$\frac{N + 1}{2} + 1$	$1 + \alpha + 1$
更新	$\frac{N + 1}{2} + 1$	$1 + \alpha + 1$

本日のまとめ

- ▶ データベースの検索のコストはディスクへの読み書きの回数(I/Oコスト)で測る。
- ▶ ヒープファイルとハッシュファイル、選択演算の条件や更新のあるなしによって向き不向きがある。

