

データベース(第9回)

情報工学科 木村昌臣



RDBMSの三大機能

1. メタデータ管理
2. 質問処理
3. トランザクション管理



1.メタデータ管理

- メタデータとは

- 「データのデータ」の意味
- RDBMSの場合、以下を指す
 - テーブルに関するデータ
 - 各テーブルの列名・属性
 - キーについてのデータ
 - ユーザー名、権限(ロール)
 - スキーマ名、スキーマ所有者、カタログ名
 - ビューに関するデータ(名前、定義、更新可能性についてのデータ)
 - アクセス法(インデックスの定義)
 - シノニム(別名)
 - 統計データ



1.メタデータ管理

- これらのメタデータをRDBMSはユーザーに提供できない
 - 質問処理(SQL)を実行するときが必要
 - 初めてそのデータベースを使うユーザーに「どんなテーブルがあるか」などの情報を提供するときが必要
- 「システムカタログ」というテーブル・ビュー群にて管理



1.メタデータ管理(情報スキーマ)

- SQL-92規格では、RDBMSはメタデータを以下の形式で提供するように定められている
 - USERS,SCHEMA,TABLES,COLUMNS・・・などの24個の**実表(DEFINITION_SCHEMA)**
 - これらが保持するメタデータをエンドユーザーやアプリケーションに提供するため、これらをSELECT文で検索するために定義された**ビュー(INFORMATION_SCHEMA)**

↑
基本的に更新不可



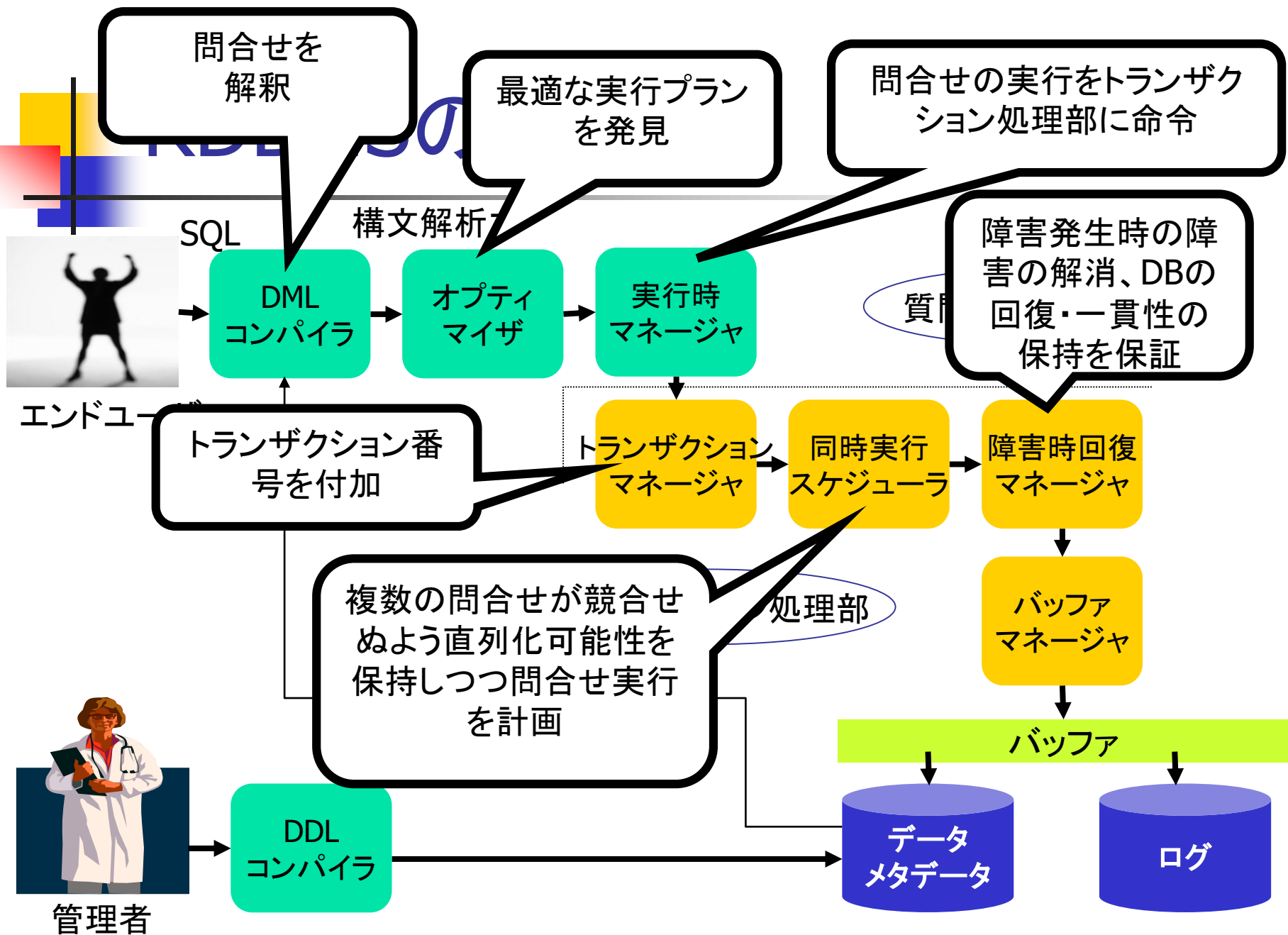
2. 質問処理

- SQLのSELECT文などの解釈・コンパイル
 - 所望のデータを検索するための内部スキーマレベルのアクセスコードを生成
- 質問処理の最適化
 - SQLは非手続き的なため、RDBMSが責任もってコードの最適化を行う必要あり



3. トランザクション管理

- トランザクション
 - データベースのデータの読み書きの単位
- これを管理することにより、データベースの一貫性を保証
 - 障害時回復
 - 同時実行制御
 - 例)書き込みに失敗したら、そのトランザクションの直前の状態まで戻す(ロールバック)



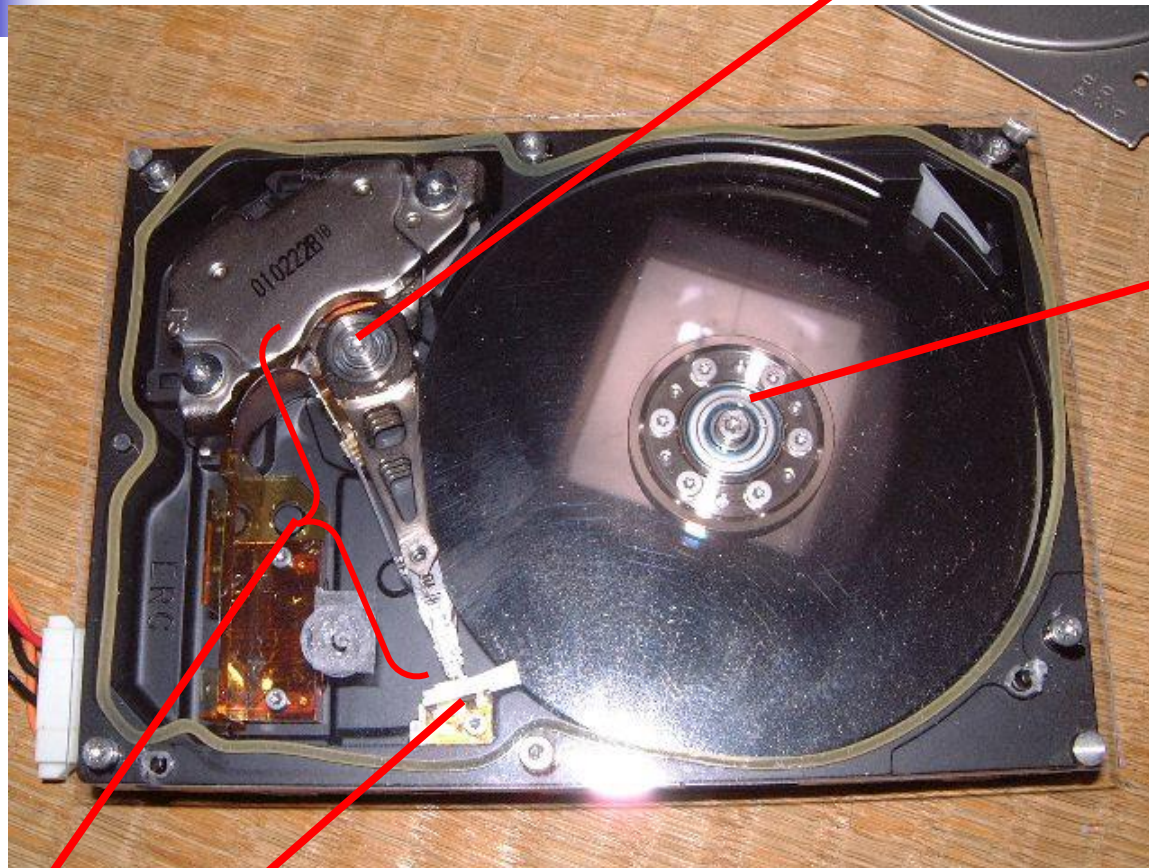
ファイルのアクセス方式と インデックス



HDDの構造

アクチュエータ

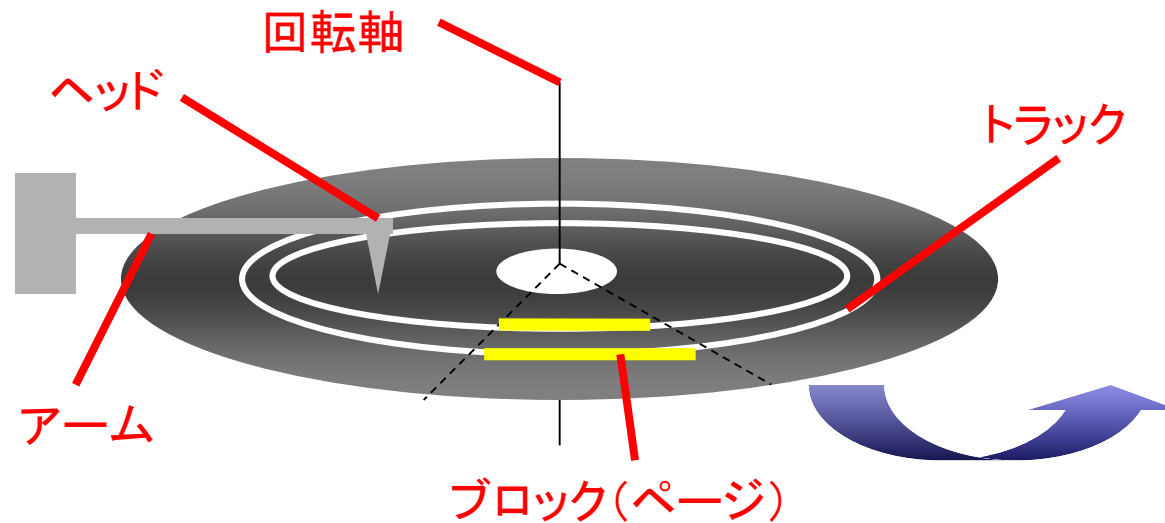
回転軸



アーム ヘッド

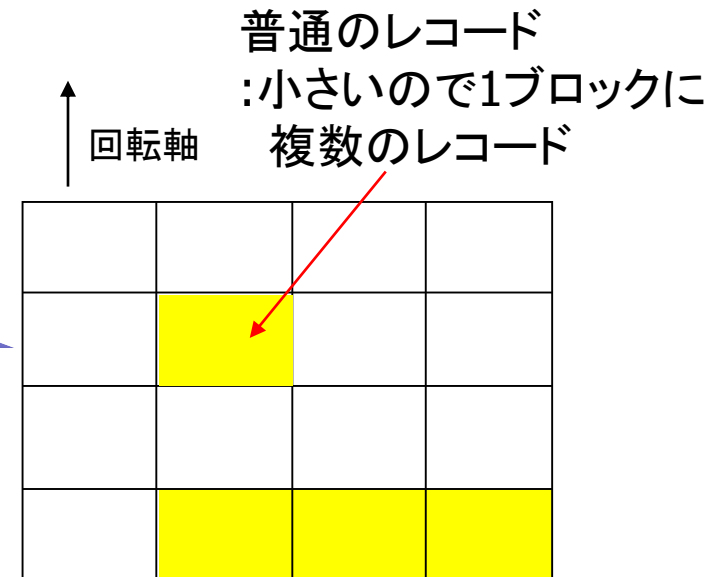
<http://www.iuec.co.jp/hdd/badsector/jpg/hdds.jpg>

HDDとファイル格納の原理



通常512バイト～4096バイト
ブロックのアドレス =

- ディスク面の認識番号
- トラック番号
- (そのトラックでの) ブロック番号の組



LOB型を含むレコード
: 大きいので複数ブロックに
1レコード

(LOB: Large Object)



ファイルアクセス方式

- ファイルの中にあるレコードを探し出す方式
 - 一つのテーブルは、ひとつのファイルとして実現されることが多いことに注意
- 代表的な例は
 - スキャン(線型探索)
 - 探索
 - インデックス法
 - ハッシュ法

スキャン

条件: 列1=8

	列1	列2
① →	10	
② →	5	
★ HIT ③ →	8	
	3	

レコードが、この順番でディスクに格納されていることが前提

ある列(**探索キー**)をもとにレコードを探索する。
探索は、テーブルの上から逐次、条件に合うか判定しながら進んでいく。

平均計算量は $O(n)$ (n:レコード数)

探索(1:2分探索)

条件: 列1=8

	列1	列2
①	1	
	3	
	4	
②	5	
	8	
③	10	

HIT

探索キー

レコードが、探索キーでソートされていることが前提

探索は、探索キーとなる列を半分ずつに区切りながら条件とマッチするまで進んでいく。

平均計算量は $O(\log_2 n)$
(n:レコード数)

探索(2:ブロック探索)

条件: 列1=8

列 1	列 2
1	
2	
4	
5	
7	
8	
25	
31	
55	

レコードが、探索キーでソートされていることが前提

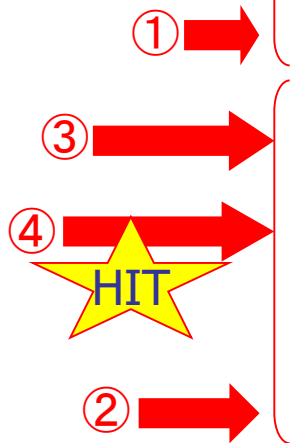
探索は、レコードをm個のブロックにわけ、まずその最後のレコードの探索キーの値と条件と比較する。

ブロックの最後のレコードの値が条件より小さければ次のブロックを探し、ブロックの最後のレコードの値が条件より大きければそのブロックをスキャンする

平均計算量は $O(\sqrt{n})$

(n:レコード数

計算量 = $n/(2m) + m/2$ を最小にするmを選ぶ)



探索キー

インデックス法

インデックスとは

テーブルのある列の値およびその値をもつレコードへのポインタの対からなるレコード群

社員マスタ

インデックスフィールド

社員番号	氏名	健保番号	部署コード
E1111	宇田川	A1111	A11
E2222	河瀬	A4055	A11
E3333	小池	A7032	B21
E4444	高橋	A4200	B21
E5555	西尾	A8099	C31

X部署コード

部署コード	ポインタ
A11	●
B21	●
C31	●

この列を
検索条件
にする場合、
この列に
インデックス
を張る

順次フ

インデックスフィールドに関して
順次か非順次かでインデックスの
内部構造が変わる

ファイル

ある列に関して順次
になっていて、かつ
その列が候補キーなら
順序キー

社員マスタ

社員番号	氏名	健保番号	部署コード
E1111	宇田川	A1111	A11
E2222	河瀬	A4055	A11
E3333	小池	A7032	B21
E4444	高橋	A4200	B21
E5555	西尾	A8099	C31

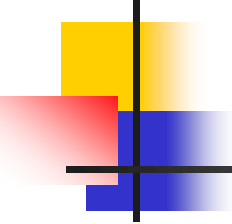
順序フィールド

社員マスタテーブル
の各レコードは
部署コードの順に
並んでいる

社員マスタテーブル
の各レコードは
健康保険番号の順に
並んでいない

社員マスタは
健保番号に関して
非順次

社員マスタは
部署コードに関して
順次



1 次インデックス・2次インデックス

■ 1次インデックス

- 順序キー上に張ったインデックス
- インデックスは元のテーブルのインデックスフィールドを順序キーとして持つ

■ 2次インデックス

- 順序キー以外の列の上に張ったインデックス
- 順序フィールドではないが候補キーである場合も1次インデックスと同様
- 候補キーでない場合は、一般に複数のレコードポイントを収めるためのブロックをはさむ。インデックスのレコードポイントは該当するブロックを指す。
- 順序フィールドだが候補キーではない場合はクラスタリングインデックスと呼ぶ

1次インデックス

■ 1次インデックス

■ 主キー上に張られたインデックス

- 主キーの値が決まればレコードは一つ決まる

■ インデックスは

- 元のテーブルの主キーの値（インデックス上のキーの並びはソートされている）
- 実際のレコードへのポインタ

の組を持ち、キーの値とレコードを直接結びつけている

社員番号	ポインタ
E1111	◆
E3333	◆
E5555	◆

社員番号	氏名	健保番号	部署コード
E1111	宇田川	A1111	A11
E2222	河瀬	A4055	A11
E3333	小池	A7032	B21
E4444	高橋	A4200	B21
E5555	西尾	A8099	C31

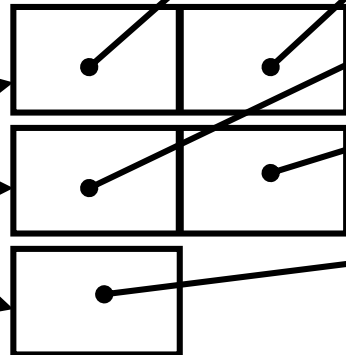
2次インデックス

■ 2次インデックス

- 順序キー以外の列の上に張ったインデックス
- 順序フィールドではないが候補キーである場合、1次インデックスと同様
- 候補キーでない場合は、一般に複数のレコードポインタを収めるためのブロックをはさむ。

X部署コード

部署コード	ポインタ
A11	
B21	
C31	



社員番号	氏名	健保番号	部署コード
E1111	宇田川	A1111	A11
E2222	河瀬	A4055	A11
E3333	小池	A7032	B21
E4444	高橋	A4200	B21
E5555	西尾	A8099	C31

クラスタリングインデックス

順序フィールドだが候補キーでない場合、その列の上で定義されたインデックスを特に「クラスタリングインデックス」と呼ぶ

X部署コード		社員マスタ			
部署コード	ポインタ	社員番号	氏名	健保番号	部署コード
A11	●	E1111	宇田川	A1111	A11
B21	●	E2222	河瀬	A4055	A11
C31	●	E3333	小池	A7032	B21
		E4444	高橋	A4200	B21
		E5555	西尾	A8099	C31

密集インデックス法

順序フィールドのすべての値をインデックス内にもつ方法

- 次の点在インデックスよりパフォーマンスはよい
- 記憶領域をよけいにくう

社員マスタ

社員番号	氏名	健保番号	部署コード
E1111	宇田川	A1111	A11
E2222	河瀬	A4055	A11
E3333	小池	A7032	B21
E4444	高橋	A4200	B21
E5555	西尾	A8099	C31

X部署コード

部署コード	ポインタ
A11	
B21	
C31	

点在インデックス法

順序フィールドの一部の値のみをインデックス内にもつ方法

- 密集インデックスよりパフォーマンスは悪い
- 記憶領域は稼げる

社員マスタ

社員番号	氏名		
E1111	宇田川		
E2222	河瀬		
E3333	小池	A7032	B21
E4444	高橋	A4200	B21
E5555	西尾	A8099	C31

C31を探すときには
B21のポインタ
が指すレコードから
たどっていく

X部署コード

部署コード	ポインタ
A11	●
B21	●

C31がない

部署コードが
C31なるレコード
を探したい

HIT