DBアクセススピードの向上技法

情報システム化技術演習－３

（データベースシステム設計）

第２回

URL　http://homepage3.nifty.com/suetsuguf/

Email　fwhy6454@mb.infoweb.ne.jp

作成者　末次文雄

復習：DB設計のポイント

DB構造図

① データ項目は重複して持たない

② 導出されるデータ項目は持たない

- 簡単な計算により容易に求まるもの
- 他のテーブルを検索すれば取り出せるもの

③ プログラム中にデータを持たない

- ﾌﾟﾛｸﾞﾗﾑ変更よりはﾃﾞｰﾀ変更のほうが容易

DB関連図

① 全てのテーブルを対象にすること

② 関連を見つけてテーブルを新設する

③ 関連の基数を求めること

目次

１．応答時間の構成

１．１　ディスクの格納方法

① ハードディスクの構造

・DBアクセス時間の大半はディスクI/O時間

DISK-----ミリセカンド単位（ms、千分の１秒）

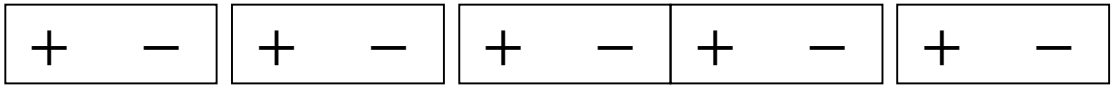
CPU------ナノセカンド単位（ns、10億分の１秒）

・呼び名　磁気ディスク、ハードディスク

（DASD-----メインフレームでの用語）

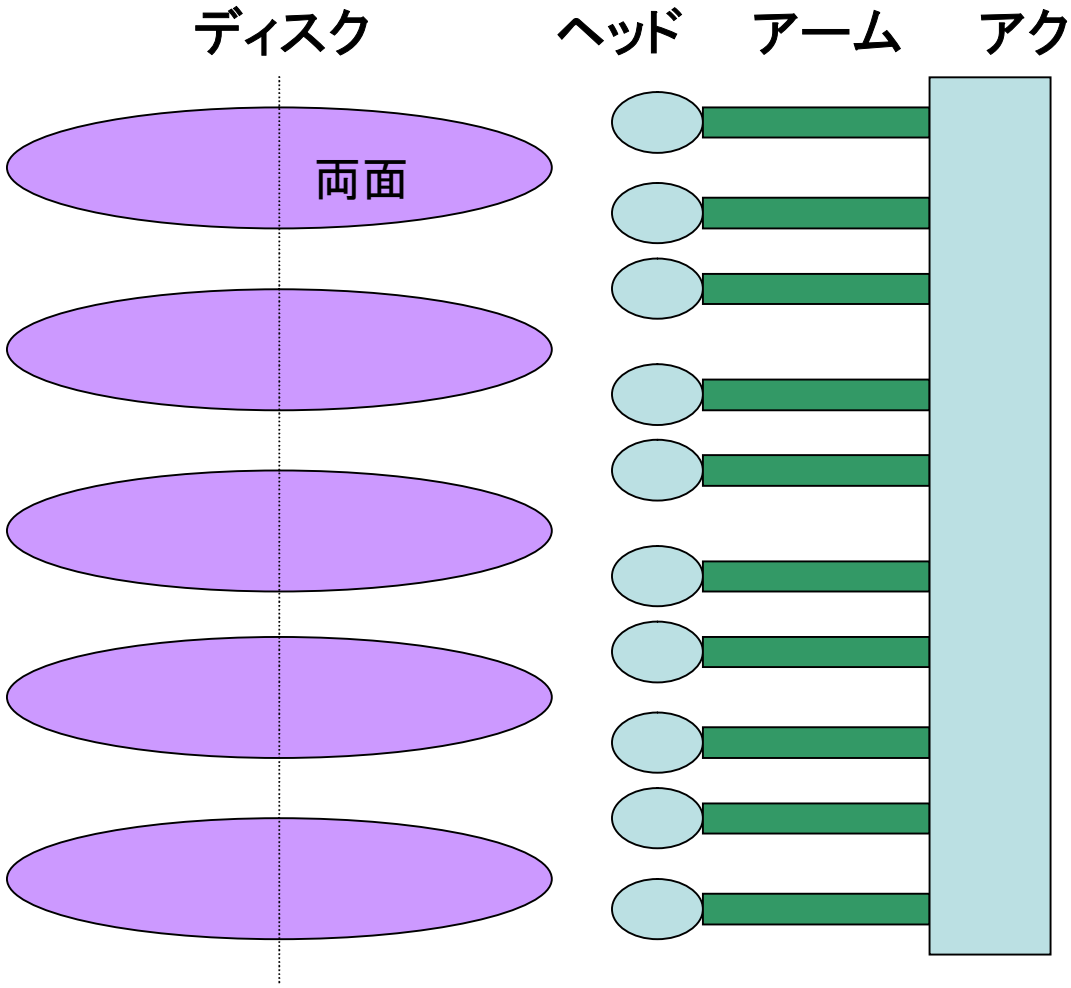
（HDD-------サーバー、パソコンでの用語）

・記録方式　　水平磁気記録



①　（続き）

- 構造



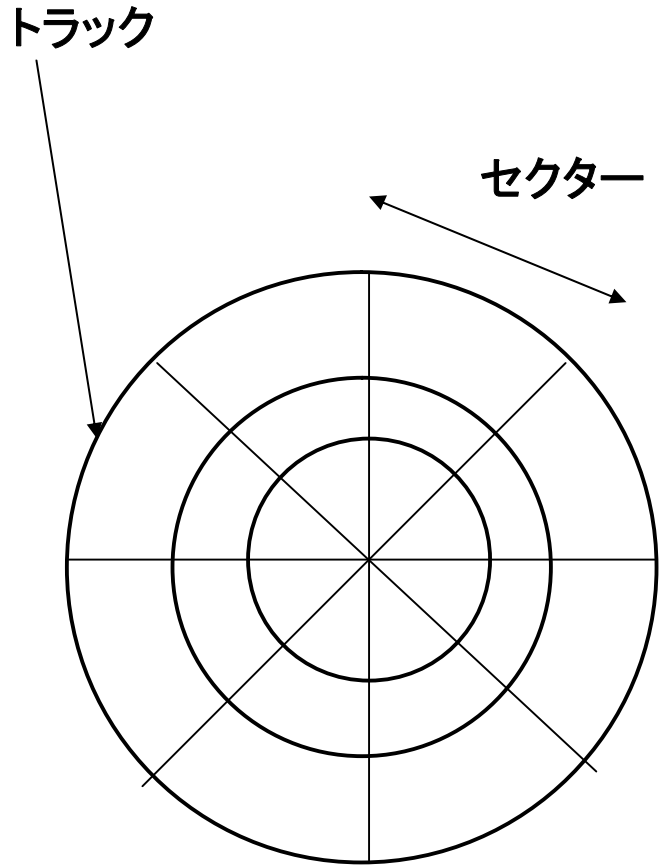
高速回転中のディスクに対して、

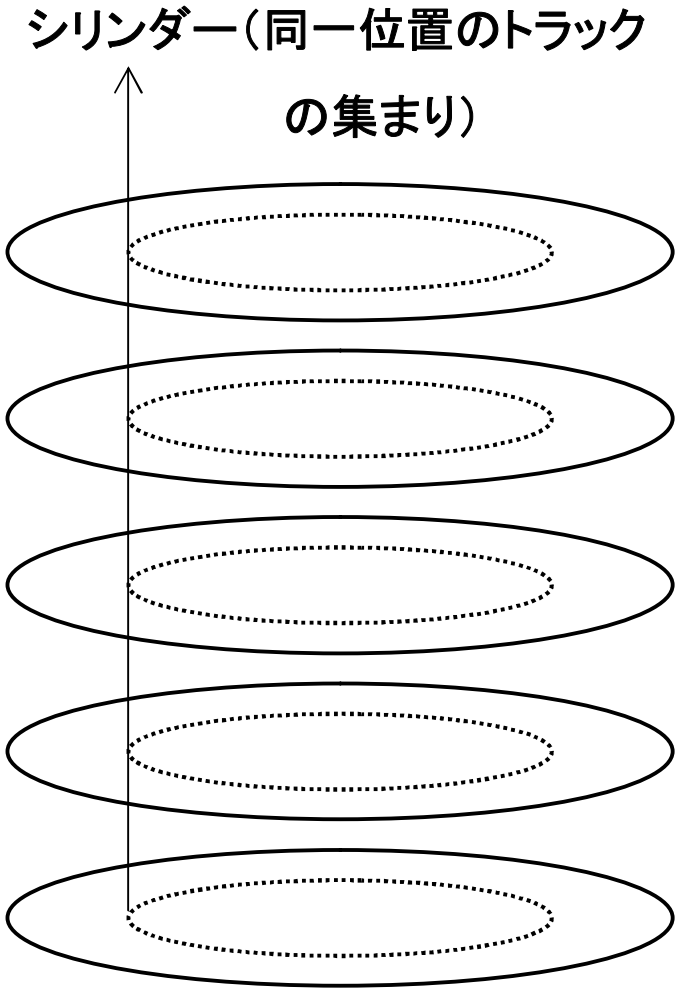
アクチュエータ単位で、全アームが前後に動き、

ヘッドをディスクの所定の位置にとめる。

①　（続き）

・ディスクの使い方





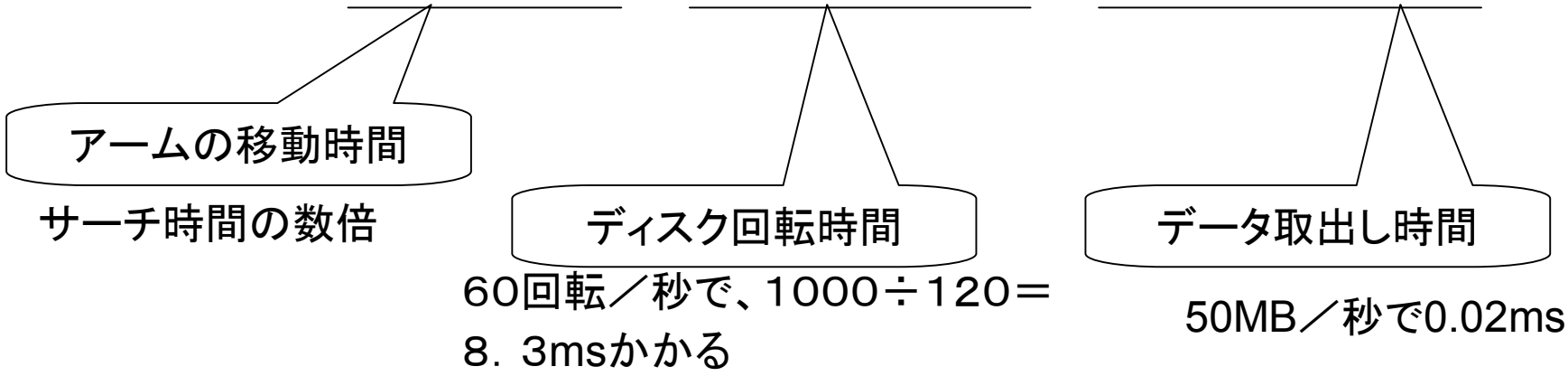
①　（続き）

- ディスクのアドレス

アドレス＝ディスク装置番号＋サーフェス番号＋トラック番号
＋セクター番号＋セクター内の相対位置番号

- ディスク検索時間

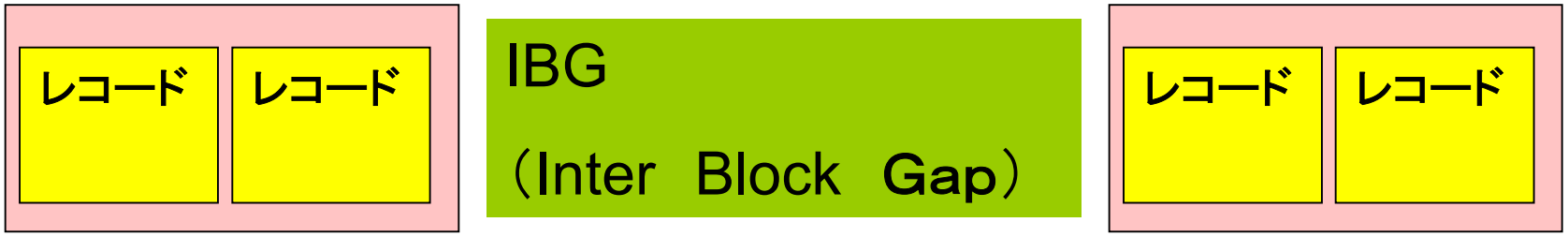
合計タイム＝シーク時間＋サーチ時間＋データ転送時間



② ブロック、ギャップ

- プログラムは、レコード単位で読み書きするが、
- ディスク装置では、読み書きの速度を上げる
  ために、ブロックという単位で読み書きを行う。
  －ブロッキング・ファクター（ブロック化因数）
  －何件かのレコードを集めて一かたまりに。

・最大は、1トラックまたは1セクター
・通常は、2K、4Kの倍数。



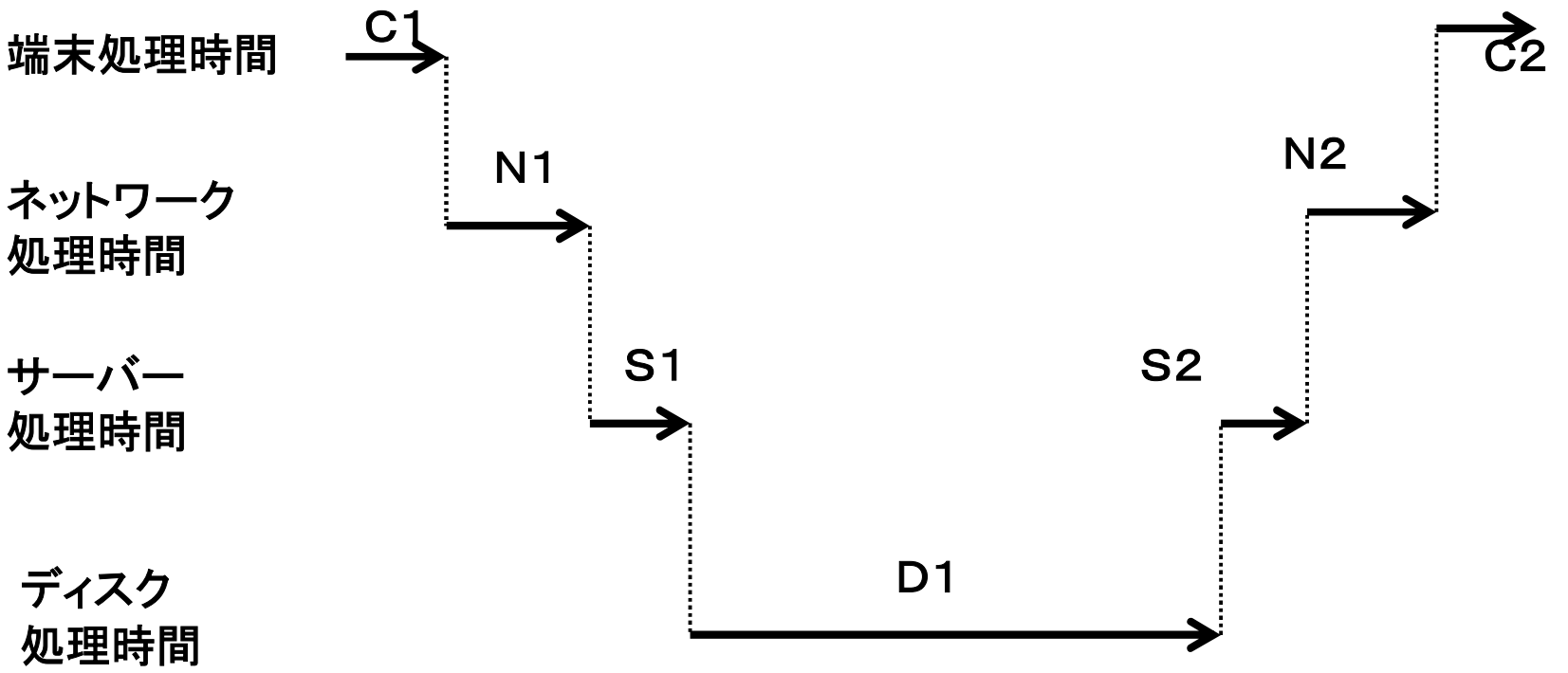
ブロック
（物理レコード）

次のIO命令までの、
ディスク回転距離

ブロック
（物理レコード）

１．２　応答時間

例：オンライン処理の場合



２．物理的なｱｸｾｽｽﾋﾟｰﾄﾞ向上策

２．１　ブロックサイズとバッファーサイズ

２．２　物理配置

２．３　インデックッスの設定

２．４　オプティマイザー

２．５　その他高速化技術

２．１　ブロックサイズとバッファーサイズ

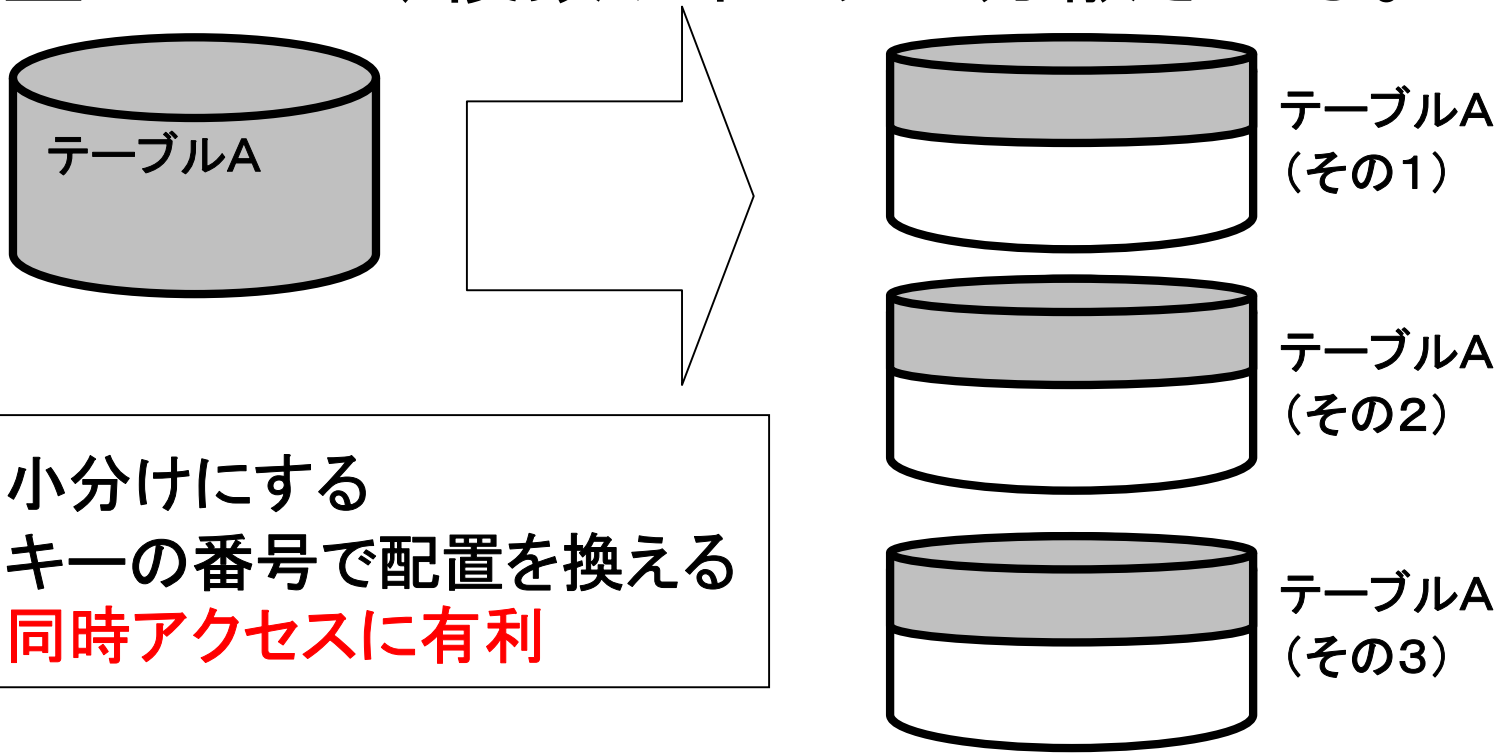
①ブロックサイズ
- ー実際のディスクの読み書きの単位
- ーレコード長の倍数を取る
- ーページとも言う
- ー通常は、DISKトラックの半分が良い

②バッファーサイズ
- ーCPUとディスク間の緩衝メモリー
- ー通常は、ブロックの倍数
- ーDBMS独自でも保有している場合が多い
- ー大きいほど良い

２．２　物理配置

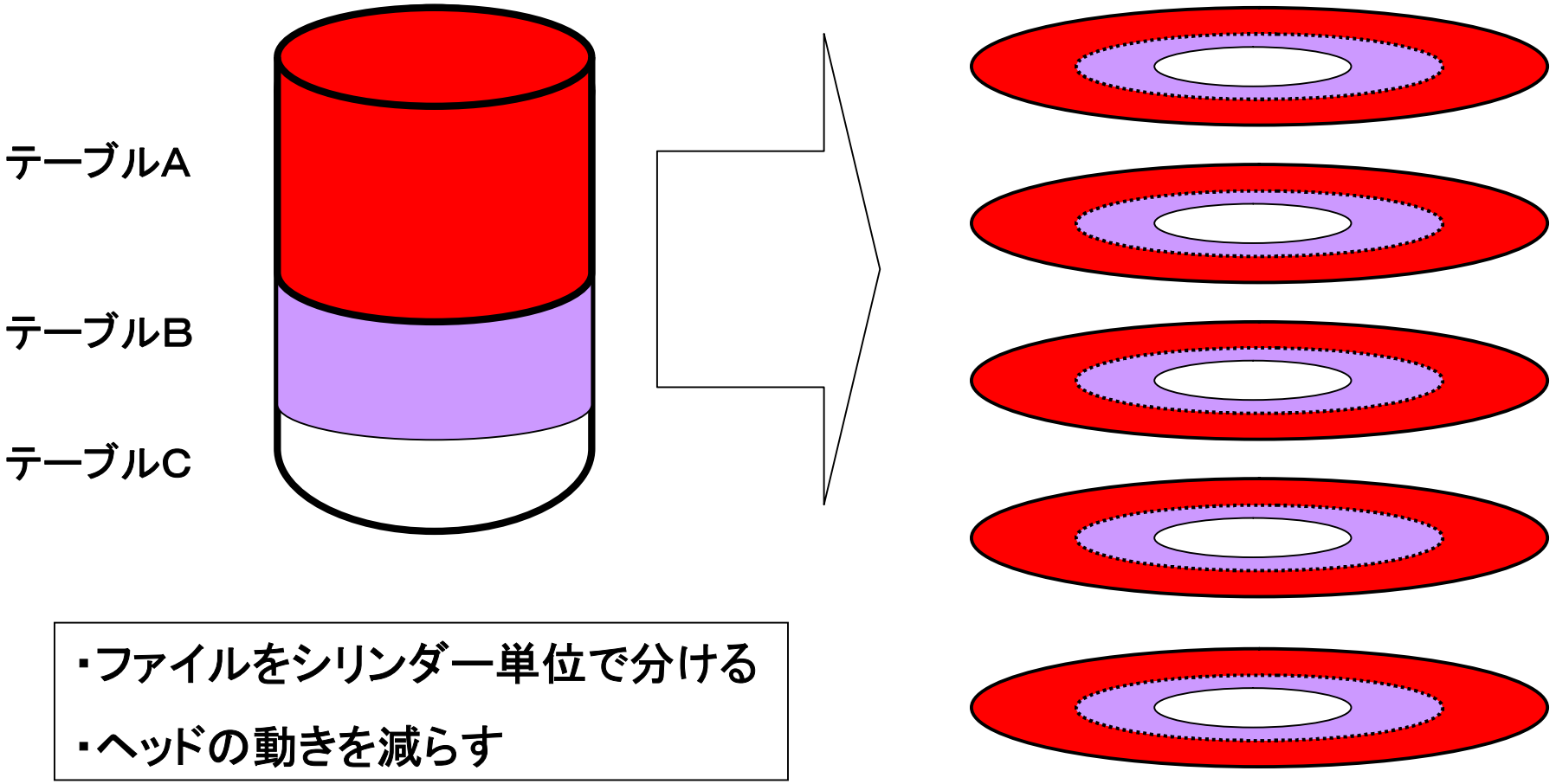
①ディスクへの分散配置

・大量DBでは、複数ディスクに分散させる。



・データ部分、インデックス部分、ログも分割。

②シリンダー別に配置



・ファイルをシリンダー単位で分ける

・ヘッドの動きを減らす

・実際には、自動的にシリンダー単位にするDBMSが多い

２．３　インデックッスの設定

- そのテーブルにアクセスする場合、
  　どのデータ項目を指定することが多いかで決める
- 複数データ項目の組合せが可能
  　ー複数データ項目中、検索頻度が高い
- テーブル定義時に、同時に設定できる
  　ーキーを最上位にすること
- DB更新時のオーバーヘッドが大きくなる。
  　ーデータだけでなく、インデックスも同時に
  　更新する
- データ量が少ないテーブルではインデックスは不要

①DBの定義

物理的なDISKの配置、
スペースを定義するもの。

```
CREATE  DATABASE  DB名
        ON（NAME=ファイル名，
            FILENAME='ﾃﾞｨﾚｸﾄﾘｰﾊﾟｽ
                        ¥ファイル名'
            SIZE=〇〇MB,
            MAXSIZE=〇〇MB,
            FILEGROWTH=〇〇%）
```

注：変更は、CREATEの代わりに、ALTERを使用。

②テーブル(表)の定義

テーブルのデータ構造を定義するもの。

CREATE　TABLE　テーブル名

（データ項目名　**データ型**(長さ)　NULLまたは
NOT　NULL
UNIQUE
DEFAULT'デフォルト値'

データ項目名
.
.　　）

③ INDEXの作成方法

単独インデックス

- INDEXの定義（1項目のケース）

CREATE 〔UNIQUE〕 INDEX インデックス名

ON 学生名簿TBL（学籍番号）

複合インデックス

- INDEXの定義（複数項目のケース）

CREATE 〔UNIQUE〕 INDEX インデックス名

ON 表の名前（項目A,項目C）

[UNIQUE] の意味： 作成されるインデックスのキーは重複が無い。
作成元のテーブルでインデックスに使うデータの重複を
チェックして、ダブリはエラーにする。

Index作成事例

- 主キーの定義方法

  CREATE　TABLE　学生名簿TBL

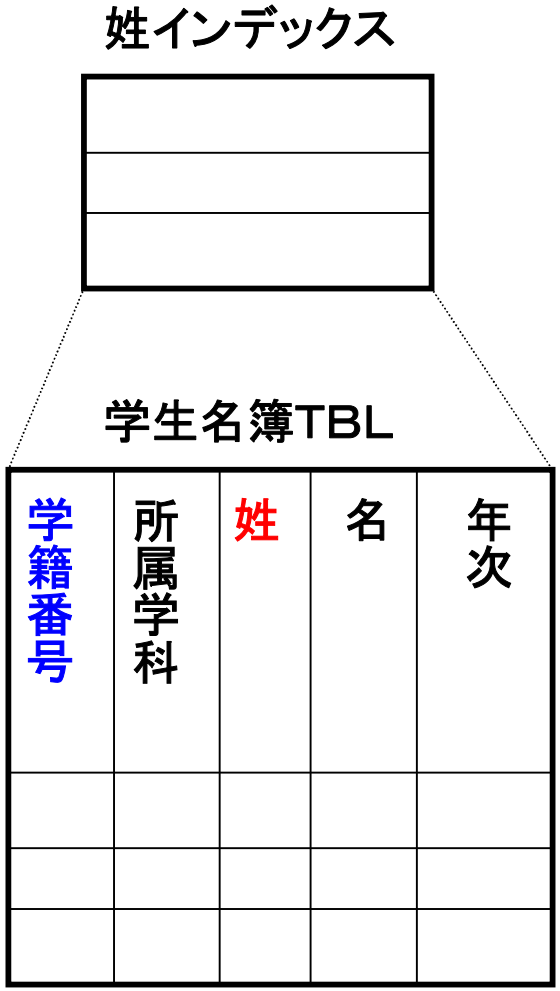
  （学籍番号　NUM（10）

  PRIMARY　KEY

- INDEX定義方法

  CREATE　INDEX　姓ｲﾝﾃﾞｯｸｽ

  ON　学生名簿TBL（姓）



（補足）テーブルにおけるキーの種類

－主キー（Primary　Key）

　　行を一意に識別するもの、複数の列指定可

－候補キー（Candidate　Key）

　　行のキー候補（社員番号と社会保険キー等）

－外部キー（External　Key）

　　他の表の主キーになっているもの

－連結キー（Concatinated　Key）

　　複数の列からなる

（補足）テーブルの定義

- 一意なデータ項目の定義（UNIQUE）---列内で一意

```
CREATE  TABLE  学生名簿TBL
    （学籍番号  NUM(10)  UNIQUE  , ・・・・・・・・・）
```

- 主キーのテーブル定義（１項目のケース）

```
CREATE  TABLE  学生名簿TBL
    （学籍番号  NUM(10)  NOT  NULL  UNIQUE
                         PRIMARY  KEY  , ・・・・・・・）
```

- 主キーのテーブル定義（複数項目のケース）

```
CREATE  TABLE  表AAAAA
    （項目A    CHAR(05)  NOT  NULL  ,
     項目B    CHAR(10)  ,
     項目C    CHAR(15)  NOT  NULL ,
     項目D    CHAR(20)  ,
    UNIQUE （項目A,項目C）,
    PRIMARY  KEY(項目A,項目C)  , ・・・・・）
```

（注）ORACLEでは、UNIQUEやPRIMARY KEY指定で、自動的にINDEX生成

＜演習問題　２．１＞

以下の課題についてインデックス作成のための
SQL文を書いてください。

課題

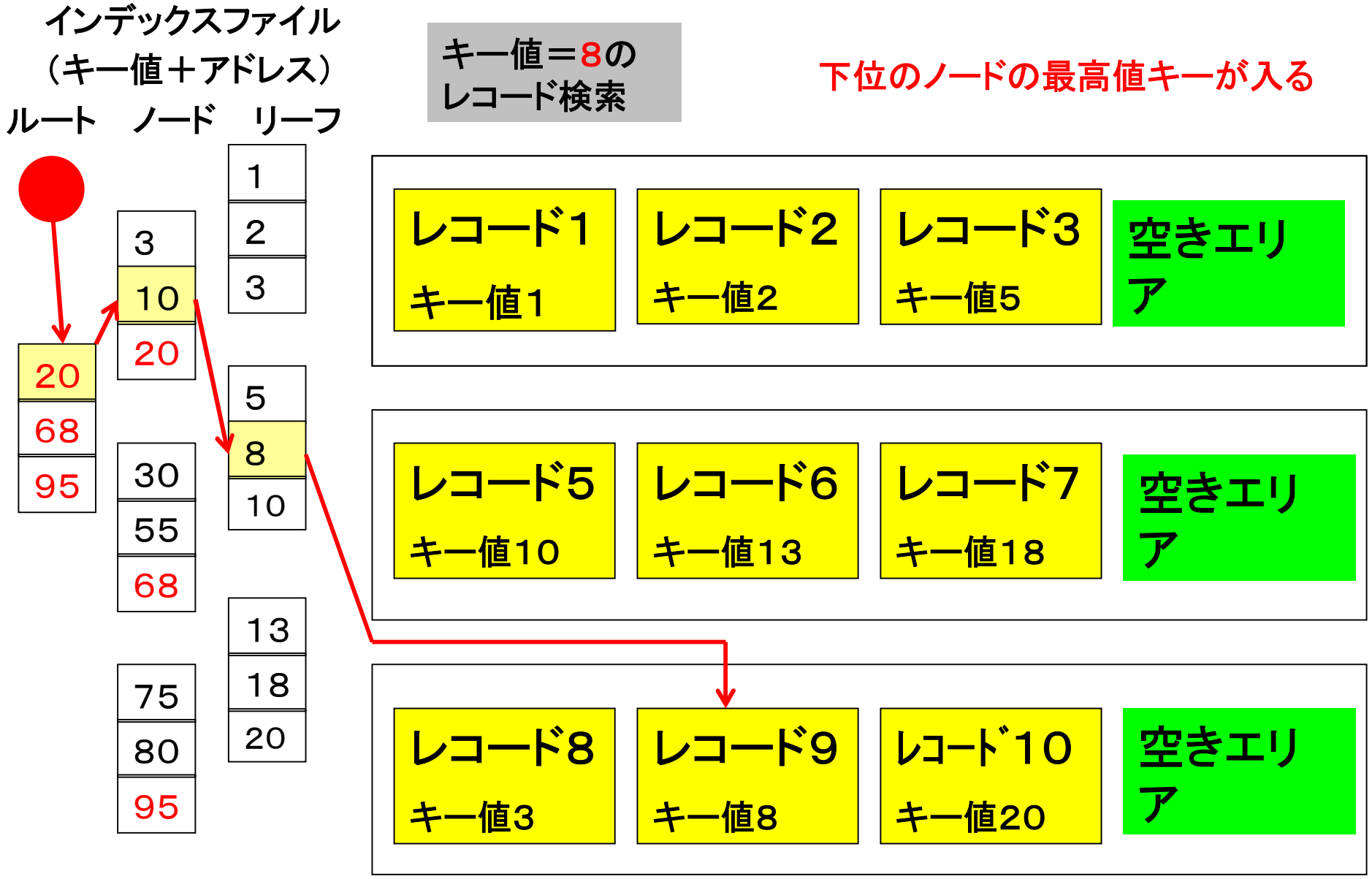
①課題１、社員属性テーブルに「社員氏名」で検索できる
　インデックスを作成せよ。

②課題２、保有スキルテーブルに「スキル分野＋スキル項目
　＋スキルレベル」でアクセスできるインデックスを作成せよ。

④　木構造索引付ファイル

- 順次編成ファイルをもとに、索引を木構造で作成する
  ことにより、大量データのキー検索を高速化したもの。
- メモリー、ディスクが急速に安価になり、
  かつCPU、ディスクの高速化により可能となった。
- 長所：順次でもキー値指定でもアクセスできる。
  レコードの取出しが高速で出来る。
- 短所：索引のための領域が増大する。
  索引の更新頻度が高まり、オーバーヘッドが大。
  データ量が少ない場合は、効果が薄い。

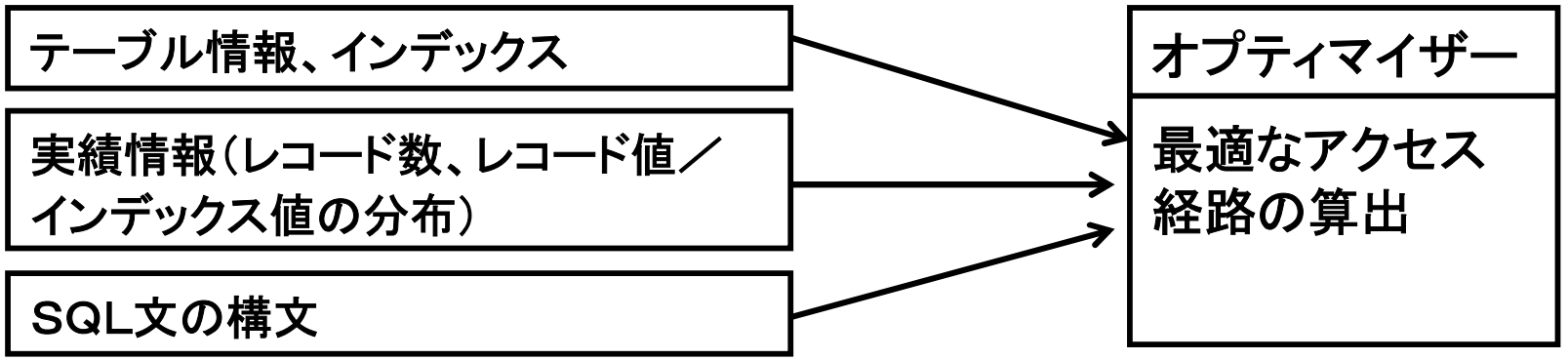
④　（続き）


インデックスファイル
（キー値＋アドレス）
ルート
ノード
リーフ
キー値＝8の
レコード検索
下位のノードの最高値キーが入る
20
68
95
3
10
20
30
55
68
75
80
95
1
2
3
5
8
10
13
18
20
レコード1
キー値1
レコード2
キー値2
レコード3
キー値5
空きエリア
レコード5
キー値10
レコード6
キー値13
レコード7
キー値18
空きエリア
レコード8
キー値3
レコード9
キー値8
レコード10
キー値20
空きエリア

２．４　オプティマイザー

①オプティマイザーとは？

- RDBMSには、検索経路の自動最適化ロジックが付いている。（最適化処理optimizer）
  - インデックスの使用、使うインデックスを決定
- SQL文の構文解析し、テーブル構造を分析することにより、SQL文の実行計画（plan）を立てて、最適なアクセス経路を検出する。



② オプティマイザーの種類

- 二つの手法が用意されている。
  - ールールベース
    - 一定の事前に定めた優先順位に従って、planを立て、アクセス経路を決定。
  - ーコストベース
    - 予め収集した情報をもとに複数のplanを立て最も安価なアクセス経路を決定。
      - （データ項目の長さ、木構造インデックスの深さ、過去の統計情報）

③ オプティマイザーの制約

- データ件数が少ない、キー値の種類が少ない場合は、インデックスを経由しない場合がある。
- SQL文が稚拙であれば、インデックスを使用せず、全件アクセスとなる場合がある。
  （例：SELECT文で、WHERE句のモレ）
- インデックスだけでなく、SQL文の作成方法も、アクセススピードに大いに関係する。

２．５　その他高速化技術

- パーティショニング（表、インデックスの分割）
- 高速ディスク（ディスクキャッシュ、RAID）
  - ・ディスクキャッシュ（一時保管のメモリ）disk　cache
  - ・RAID（組合せディスク）Redundant Array of Inexpensive Disks
- DBサーバー（DBMSのみ稼動させるサーバー）
- 並列処理（但しパーティショニングが前提）
- ストレッジ階層化（半導体ディスク＋RAID）
- メインメモリDB（大容量メモリー）

３．論理DB構造面からのｱｸｾｽｽﾋﾟｰﾄﾞ向上

３．１　逆正規化

- 頻繁に複数のテーブルを参照する場合は、
  ディスクアクセス回数を減らすために、
  あえて正規化をしない場合がある。
  - ・繰り返しデータを持つ（時系列データなど）

営業店	店名	1Q売上	2Q売上	3Q売上	4Q売上

  - ・データを重複させる

社員番号	社員氏名	支給年月	支給金額

- ただし、データ更新は2重になる。

３．２　導出データを持つ

- 頻繁にアクセス対象になる場合は、計算処理時間を短縮するため、導出データといえどもデータとして持つ場合がある。
  - ー合計値など（小計、中計、総合計、計算値・・・）
  - ー課題１の給与合計、変動給与・・・
  - ー他のDBの検索結果を持つ
- ただし、リアルタイムで導出データが変わる場合は、導出データの更新というオーバーヘッドがかかり不利になる。

＜演習問題　２．２＞

アクセススピード向上策として、以下の課題について
DB構造の見直しをしてください。

課題

① 給与計算用DBについて導出データを持つDB構造を考えよ。

４．実用的なアクセススピード向上策

４．１　開発前に実量での検証

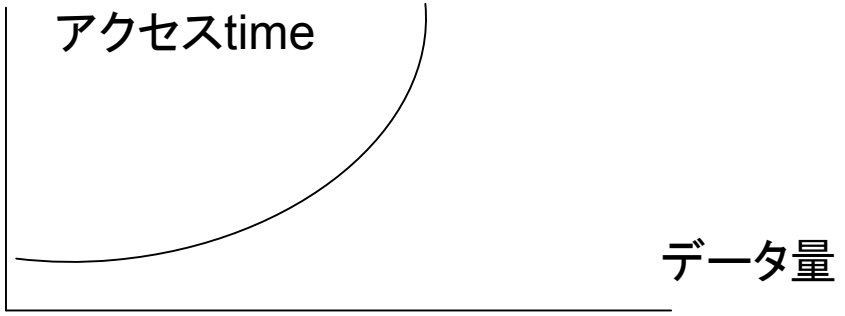
４．２　不要データの削除

４．３　DB分割

４．４　コピーDB作成

注：DASDアクセススピードは、データ量が増えれば、急速に悪化する

悪化の傾向は、
・データ量
・DASDの種類
・DBMS　で異なる



４．１　開発前に実量での検証

- 設計前に、HW、SWの検証をする
- 実環境に近いこと
  - ・HW、SWとも準備
  - ・一時貸借ででも実施する
- 実際のデータ量でのDB生成
- DB更新、DB検索とも検証用プログラム開発
  - ・オプティマイザー含めた検証
- その結果で、システム開発期間中に対応策実施
  - ・DB構造、インデックス、DBMSの検証

４．２　不要データの削除

- データの保管要否を検討（要否、保管期間）
- 移行時に現行システムのデータを削減する
  - －現行データの全件検証が必要
  - －不要データを見つけ出す
- 運用時には、不要な履歴データを削除
  - －定期的にDBからデータ削除
  - －直後に、DB再編成（リロード）

・データ量の増加により、DBアクセスは、
ある量から急激に悪化する。

４．３　DB分割

- 大量データは、DB分割をする
- 分割方法は二つ

① カレントデータ、ヒストリカルデータに分割

ーカレント　　＝通常の業務で使用するデータ

ーヒストリカル＝過去のデータであり、使用機会が極端に少ないデータ、更新しないもの

（例示）旧商品データ、退職者データ

② 種類別に分割

ー業務上、意味のある分割（商品大分類、分野別・・・）

ーDB更新、DB検索システムは派生型で開発できる

４．４　コピーDB作成

- 検索量が大きなDBは、検索用コピーを作成
  - －更新用、検索用DBを持つ
  - －夜間に、更新用DBから検索用DBをコピー
- 全件データの順次検索用に、コピーDBを作成
  - －利用サイクルに合わせて作成
  - －夜間に、更新用DBから順次用DBをコピー
  - －ロジカルバックアップとも言う

＜演習問題　２．３＞

アクセススピード向上策として、以下の課題について
DB分割した結果のDB構造図を作成してください。

課題

①人材派遣用DBについてカレント、ヒストリカルにDB分割する。