1. マルチメディアデータベース序論

（マルチメディアデータベース序論，全６回）

https://www.kkaneko.jp/de/multimediadb/index.html

金子邦彦

Database System が必要となる理由

- データの共有が簡単になる

- 無駄が減る

データを共有せずに，プログラムごとにファイルを持つ場合と比べて.

- 「トランザクション」の機能

銀行システム（振込み，現金引き出しなどの機能）などのトランザクションを，確実に実行するための機能

メディア型

- 「古典的」なデータ
  - ファイル
  - Relation
  - オブジェクト
- メディア型
  - テキスト／文書　(Text/Document）
  - 静止画像　（Image)
  - 動画像　(Video)
  - 音声　(Audio)

マルチメディアデータベースシステムに期待される機能

- さまざまなメディア型

- マルチメディアを簡単に扱えるようにするための仕掛け

マルチメディア検索の特質

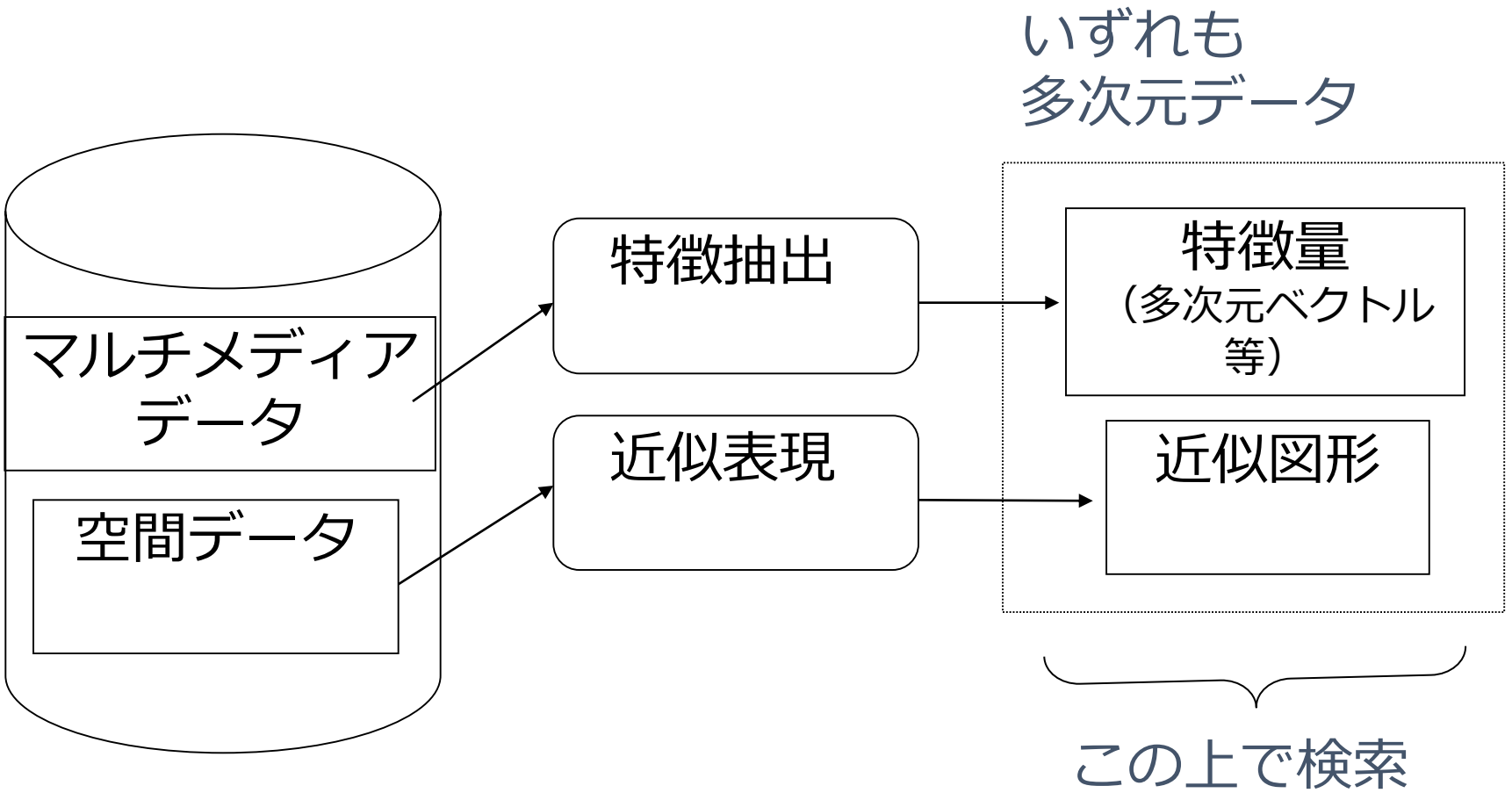
- 内容検索　／　属性による検索
  - 条件による検索
  - 「例示」による検索

} マルチメディアでは意味がある

- 類似検索　／　一致検索
  - 等しい
  - より大きい
  - より小さい
  - 「似ている」
  - など

} マルチメディアでは意味がある

マルチメディアデータベースの枠組み



マルチメディアの例

- 犯罪捜査のためのデータベース
  - 動画像：　監視カメラ、車載カメラで得る
  - 静止画像：　調査員が持つカメラで得る
  - 文書：　別途付加された情報
  - Relational data：　人口、住所、その他の背景情報
  - Geographic data：道、川、行政界、建物など座標に関わる情報

オペレーション

- 入力　　形状モデリング
- 出力　　仮想現実表示　多重解像度表現, プログレッシブメッシュ(Hoppe), 再分割接続性
- 編集
- 全件検索
- 索引付加
- ハイパーリンク付加
- Content 抽出と類似検索

マルチメディアデータベースにおける質問処理

- メディア依存の「属性」と「オペレーション」
  - 属性とオペレーションはメディア依存であり、質問処理もメディア依存にならねばならない
- 検索条件の多様さ
  - 検索条件として，データの属性だけでなく、内容（Content)や関連（Relationship)による検索が行われる
- あいまい検索
  - 「類似度（Similarity)」によるあいまい検索が重要になる

画像データの性質

- 画像には、「もの」が写っている
- 画像データの属性は：
  - 形
  - 場所
  - 色、明るさ
    - 色や明るさは、「画素」が持つ性質
    - しかし、「画素」は、直接扱うのは手間がかかる
    - 「ものが写っている長方形領域（セル）」を単位として扱うと、扱いやすい

画像データベースの問い合わせ例（１／２）

- ある捜査員　X氏は、Y氏の今までの写真を調べたいと考えた

[問い合わせの例]

- 　データベースの静止画像から、「Y氏」が写っている全ての画像を得よ

→　名前（文字列）による検索

画像データベースの問い合わせ例（２／２）

- ある捜査員　X氏は、ある人の顔写真を撮った
- X氏は、その写真が誰かを知りたいと考えた

[問い合わせの例]

- 　データベースの静止画像から、この顔写真の人が写っている全ての画像を得よ。

　　→　例示画像による検索

画像データベースにおける「あいまい検索」

- 普通のデータベース
  - ユーザは、「問い合わせ」を入力し、答えとして検索結果を得る

- 画像データベース
  - ユーザの手元に、１枚の写真（例示画像）があって、「この写真に写っている人に似ている写真」を、データベースの中から探す
  - 普通のデータベースとは異なる
  - システムは、入力画像（例示画像）とデータベース内の画像とのマッチングを行う

あいまい検索

- あいまい検索

  “Similar to” のような句が導入される

```
select p
from p in Pictures
where p.similar_to example1
```

- 重み
  - 各検索条件は、重要度に応じて重み付けされる

  where “株式” １００% “破綻” ９０% “信用秩序” ５０%

ビデオデータの特質

- フレーム
  - あるフレームに，登場人物，登場物，振る舞いが写っている
- 登場人物，登場物
  - 登場人物，登場物には，「属性」がある
    （例えば，名前，年齢など）
- 振る舞い
  - 振る舞いにも，「属性」がある
    （例えば，歌の名前，誰に話し掛けているかなど）
    「属性」は，刻々と変わる

ビデオデータへの
OQL問い合わせの例

```
SELECT    v[start, end]
FROM      Videos v
WHERE     v.segment(start, end) IN
                    v.FindVideoWithObject(Dennis)
```

時間的性質（上記の例では「start, end」）や
写っている「もの」の中身が考慮されねばならない

データベース

永続データ
データの共有
データモデル
データ検索言語，データ操作言語
「トランザクション」の機能

関係データベース

関係モデル
関係代数
Relational Calculus

SQL

オブジェクトデータベース

オブジェクト指向

OQL
OML

関係データベース

- 関係データベースのデータの単位： テーブル
  表の形式

- テーブルの各行：　タップル（tuple）
  １つのデータのまとまり

- テーブルの各列：
  同じタイプのデータ（属性）が並んでいる.

関係データベースの例

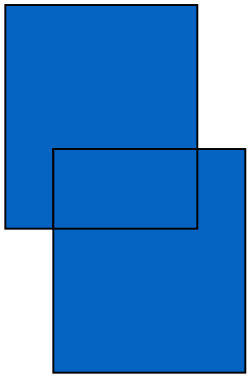
スキーマ例

(COMP, SSN, FNAME, LNAME, STREETNUM, STREETNAME, CITY, STATE, ZIP)

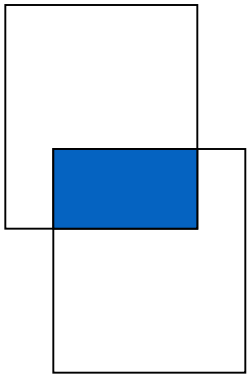
COMP	SSN	FNAME	LNAME	STREETNUM	STREETNAME	CITY	STATE	ZIP
ABC Corp.	99278	John	Simth	27	Canal St.	Fairfax	VA	22087
ABC Corp.	28745	Denia	Jones	786	Baker St.	Manassas	VA	22185

関係代数のオペレータ（２／３）

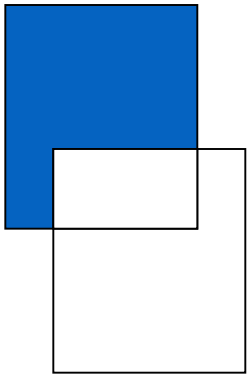
Union



Intersection



Difference



Cartesian Product

a1	b1
a2	b2

c1	d1
c2	d2

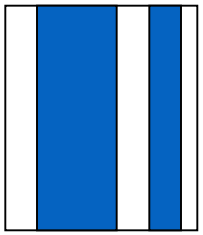
→

a1	b1	c1	d1
a2	b2	c1	d1
a1	b1	c2	d2
a2	v2	c2	d2

関係代数のオペレータ（３／３）

Selection

Projection



Natural (Join)

a1	b1
a2	b2

b1	e1
b2	e2

→

a1	b1	e1
a2	b2	e2

Divide

a
b
c

a	x
a	y
a	z
b	x

x
z

→

x

SQL

- 典型的なＳＱＬ文の例

```
SELECT    attr1, attr2, … , attrn
FROM      R1<V1>, R2<V2>, … , Rk<Vk>
<WHERE    F>
```

- R1, R2, …, Rk: テーブル名
- V1, V2, … , Vk：　　タップル変数（省略してもよい）
- WHERE句は、省略してもよい

ＳＱＬでの更新，挿入，削除

- テーブル　社員（氏名，自宅番号）に対して
- 挿入

ＩＮＳＥＲＴ　ＩＮＴＯ　社員（氏名，自宅番号）

ＶＡＬＵＥＳ（“金子邦彦”，“092-642-4XXX”）

- 更新

ＵＰＤＡＴＥ　社員

ＳＥＴ　自宅番号＝“092-643-0XXX”

ＷＨＥＲＥ　氏名＝“金子邦彦”

- 削除

ＤＥＬＥＴＥ　ＦＲＯＭ　社員

ＷＨＥＲＥ　氏名＝“金子邦彦”

オブジェクトデータベース

- 属性として，「構造をもったデータ」を扱える

  record, set, list など

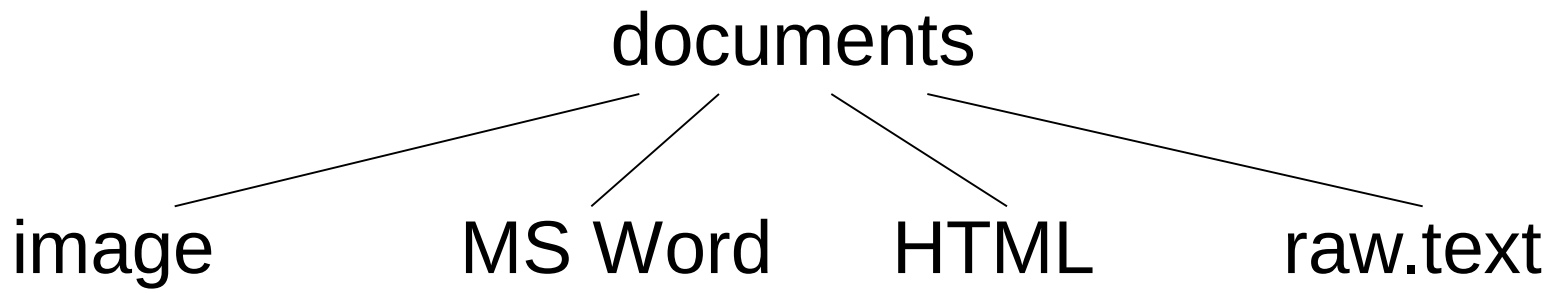
- メソッド

  クラスごとにメソッドを定義可能

  OQLから呼び出し可能

クラス階層(1/2)

- クラスとは：　同じタイプのデータの集まり



オブジェクトの属性

- 各々のオブジェクトは，属性を持つ
  - author: 文字列
  - date-created: 日付
  - admission-fee: 0 以上の浮動小数点数

- 同じクラスのオブジェクトは，同じ属性を持つ

オブジェクト

- オブジェクトには，固有の番号がついている

- オブジェクトは，属性値の並びを含む

オブジェクトＩＤ	属性値の並び

オブジェクトの宣言(1/2)

- データベースに格納すべきオブジェクトは，属性値を使って，宣言されねばならない

```
declare     b2
values      [author = John Smith,
                    URL = http://www.somewhere
                    date-created = (15, Nov, 1996 ) ]
```

オブジェクトの宣言(2/2)

```
declare     b3
values      [author = John Smith,
                    URL = http://www.somewhere
                    date-created = (15, Nov, 1996 )
                    {[link = b1], [link = b2]} ]
```

データ型

- オブジェクトの属性は型付けされている
  - 単純型

    float, int, bool, string など

  - 構造をもった型

    record, set, list など

データ型の例

- author: string
- URL: utltype
- date-created: date
- Links: Set<ObjectID>

クラスの親子関係の例

- Museum クラス

  address: string

  director: string

  departments: Set<string>

- ArtMuseum クラス

  address: string

  director: string

  departments: Set<string>

  old-master-collection: Set<string>

  modern-art-collection: Set<string>

オブジェクト定義言語
(Object Definition Language)

- オブジェクトデータベースのスキーマ：
  - オブジェクトの持つ属性　（名前と型）
  - メソッド　　（名前，引数，返り値）

- オブジェクト定義言語は，オブジェクトデータベースのスキーマを記述するための言語

ODL の例

```
interface html: documents
        { extent      html_documents
                      { attribute string author;
                        attribute date date_created;
                        relationship Set<Person> author
                              inverse Person:written_work;
                      }
}
```

ODL

interface html: documents

クラス名は　html

親クラスは　documents

{ extent html_documents

htmlクラスのオブジェクトの集まりに

「html_documents」という名前を付けよ

という意味

オブジェクト問い合わせ言語 (Object Query Language)

- SQL

  関係データベースのための問い合わせ言語

  「テーブル」，「タップル」を扱う

- ＯＱＬ

  オブジェクトデータベースのための問い合わせ言語

  「クラス」，「オブジェクト」を扱う

  ＳＱＬの上位互換

ＯＱＬの例 (1/2)

```
SELECT    struct(field1:x.url, field2:x.link)
FROM      HTMLs x
WHERE     x.author = “John.Smith”
```

HTMLクラスのオブジェクトについて，

author が “John.Smith” であるようなオブジェクトを検索せよ

ＯＱＬの例　(2/2)

```
SELECT    y.author
FROM
              SELECT    struct(field1:x.url, field2:x.link)
              FROM      MSWord x
              WHERE     x.author = “John.Smith”　）y
```

OQL は，「入れ子」になることがある

おわりに

- オブジェクトデータベースは「さまざまな種類」のメディアを扱うデータベースの「核」となりえる
- さまざまな技術
  - 質問処理（メディア依存）
  - 検索条件の多様さ（内容，関連）
  - あいまい検索