

具体的な講義の項目

- イン트로ダクション
- データベースのモデル
- 関係データベースの操作(関係代数、関係論理)
- 正規化、**概念設計**
- **問い合わせ言語**
- トランザクション処理
- データベースの内部構造
- データベースの問い合わせ処理
- その他(オブジェクト指向DB、XML/RDF-DB等々)

2020/5/7

データベース (©横田)

110

データベースの概念設計

- 実体関連モデル [Peter P. Chen, 1976]
 - ER-Model (Entity/Relationship Model)
 - データベースで表現すべき実世界の対象をモデル化
 - モデルをベースにデータベースを設計
 - 有用な意味概念の特定
 - 対応する記号オブジェクトの考察
- 正規化とERモデルの関係
 - 正規化: 大きな関係から小さな関係への写像
 - ERモデル: 実世界を示す大きな関係に到達する手段
 - 相補的

2020/5/14

データベース (©横田)

111

実体関連モデル

- 実体、関連、特性、および関連図でモデル化
 - 実体 (Entity): 明確に見極められる対象
 - 例: 商品、販売員、製造会社、販売地域
 - 関連 (Relationship): 実体間の連携
 - 例: 販売員と商品の間の販売実績、商品と製造会社の間の製造元、販売員と販売地域の間の所属
 - 実体の特別な種類と見なす
 - 全く同じオブジェクトを実体とも関連とも見なすことが可能
 - 特性 (Property): 実体あるいは関連の持つ性質
 - 例: 商品名、商品番号、販売員名、販売価格
 - 関係データベースの属性名に対応

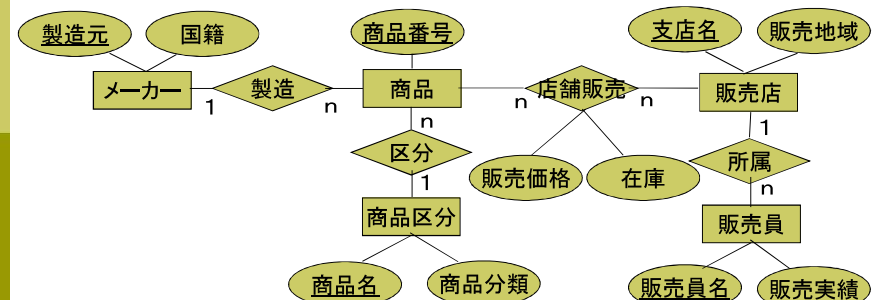
2020/5/14

データベース (©横田)

112

実体関連図 (Entity Relationship Diagram)

- 対象実世界を視覚的に表現するための図式
 - 実体は長方形、関連は菱形、特性は楕円で表現
 - 実体と関連およびそれらの特性の間を実線で結ぶ
- 例(前出の関係とは多少食い違いあり):



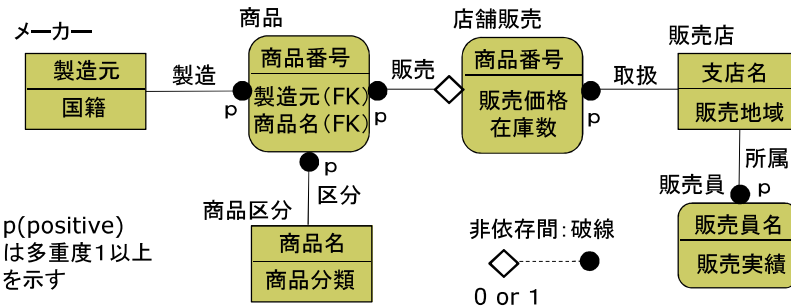
2020/5/14

データベース (©横田)

113

実体関連図: 別の記法

- IDEF1X (米国立標準技術研究所NIST作成)
 - 実体を依存実体(他の実体に依存)と非依存実体で分類
 - 非依存実体(他の実体があっても存在)を四角、依存実体を丸四角で表現
 - 実体の中に特性を記入(主キーを一番上)、関連を線と丸・菱形
- 例:



2020/5/14

データベース (©横田)

114

ちょっと考えてみよう(4-1)

- 航空便の座席の予約のためのデータベースを想定したとき、どんな実体と関連と特性が考えられるか、実体名、関連名、特性名で思いつくものを挙げてみましょう。図の入力は無理なので、
 - 実体名: ○○○○、○○○○、...
 - 関連名: ○○○○、○○○○、...
 - 特性名: ○○○○、○○○○、...
- のような形で書いてみましょう。

2020/5/14

データベース (©横田)

115

データベース言語 (Database Language)

- 2つの分類法
 - 利用対象: 定義と操作
 - 利用形態: 直接利用とプログラムからの利用
- 定義と操作
 - データ定義言語 (DDL: Data Definition Language)
 - データベース管理者 (Database Administrator) 用
 - データ操作言語 (DML: Data Manipulation Language)
 - 一般ユーザの検索/更新用

2020/5/14

データベース (©横田)

116

データベース言語 (続き)

- 利用形態による分類
 - 対話的操作
 - 端末からオンラインで DDL/DML を使い、定義、検索、更新を行なう
 - 親言語 (Host Language) とデータ副言語 (Data Sublanguage)
 - C で記述されたアプリケーションプログラムの中からデータ副言語を使う
 - プリコンパイラで DDL/DML を抽出
 - データベースとのやり取りを親言語の変数にバインド

2020/5/14

データベース (©横田)

117

問い合わせ言語の持つべき性質

- 関係完備性 (Relational Completeness)
 - by E. F. Codd [1972]
 - 与えられた関係の集まりから、(タプル・領域)関係論理式もしくは関係代数式によって得られる関係が、ある言語を使用することによって検索できる。
 - 以降説明する SQL、QBE、Que1 は、関係完備である。

ちょっと考えてみよう(4-2)

- 関係完備性を満たさないということは、その言語では正規形の関係扱えない可能性があるということになる。それはなぜでしょうか？

問い合わせ言語の例:SQL

- System R Project (1979-84)
 - IBM サンノゼ研究所 (San Jose Research Laboratory)
 - 現 Almaden Research Center
 - SEQUEL: Structured English Query Language
 - by D. D. Chamberlin, et al.
 - → SQL (Structured Query Language)
 - 標準化 (ANSI / ISO / JIS): SQL2(SQL-92) → SQL3

SQLについて

- 機能:
 - DML + DDL (+ DCL)
 - 検索、統計(集約)演算 (Aggregate Function)、更新
 - 定義、VIEW 定義
- 利用形態:
 - 対話型 (Direct SQL) / 埋め込み型 (Embedded SQL)
- 基本構文
 - SELECT 属性名の並び
 - FROM 関係名の並び
 - WHERE 検索条件 ;

その他の問い合わせ言語の例

□ QBE (Query By Example)

- IBM Yorktown Heights (T.J. Watson 研究所)
 - by H. M. Zloof, et al.
- 二次元構文による対話的インタフェース
 - (関係論理をベース)

□ Quel(関係論理をベース)

- INGRES
 - by Stonbraker (UCB → MIT)
 - → Postgres (Post ingres) → PostgreSQL
 - MS SQL Server の基

2020/5/14

データベース (©横田)

122

簡単な検索の例(1)

SQL

QBE

射影(デルタ射影)

SELECT 商品名
FROM 製造元;

製造元	商品番号	商品名	製造元
		P.ALL	

射影

SELECT DISTINCT 商品名
FROM 製造元;

製造元	商品番号	商品名	製造元
		P.	

2020/5/14

データベース (©横田)

123

簡単な検索の例(2)

SQL

QBE

選択

SELECT 商品番号、商品名、製造元
FROM 製造元
WHERE 製造元="Hanon";

製造元	商品番号	商品名	製造元
	P.	P.	Hanon

θ-選択

SELECT *
FROM 秋葉原支店
WHERE 在庫 > 10;

秋葉原支店	商品番号	販売価格	在庫
P.			> 10

2020/5/14

データベース (©横田)

124

簡単な検索の例(3)

SQL

QBE

AND条件

SELECT *
FROM 秋葉原支店
WHERE 在庫 > 10
AND 販売価格 ≤ 100,000;

秋葉原支店	商品番号	販売価格	在庫
P.		≤ 100,000	> 10

OR条件

SELECT *
FROM 秋葉原支店
WHERE 在庫 > 10
OR 販売価格 ≤ 100,000 ;

秋葉原支店	商品番号	販売価格	在庫
P.X P.Y		≤ 100,000	> 10

2020/5/14

データベース (©横田)

125

集合演算の例(1)

SQL

```
SELECT *
FROM 秋葉原支店
UNION
SELECT *
FROM 日本橋支店;
```

QBE

集合和

秋葉原支店	商品番号	販売価格	在庫
X			
日本橋支店	商品番号	販売価格	在庫
Y			
	商品番号	販売価格	在庫
P.X			
P.Y			

2020/5/14

データベース (©横田)

126

集合演算の例(2)

```
SELECT *
FROM 秋葉原支店
INTERSECTION
SELECT *
FROM 日本橋支店;
```

集合積

秋葉原支店	商品番号	販売価格	在庫
P.X			
日本橋支店	商品番号	販売価格	在庫
X			

集合差

```
SELECT *
FROM 秋葉原支店
EXCEPT
SELECT *
FROM 日本橋支店;
```

秋葉原支店	商品番号	販売価格	在庫
P. ≠ X			
日本橋支店	商品番号	販売価格	在庫
X			

2020/5/14

データベース (©横田)

127

ちょっと考えてみよう(4-3)

□ ここまで説明した SQL で、次の問い合わせは記述かのです。SQLで書いてみましょう。

- 秋葉原支店か日本橋支店で、販売価格が100,000円以上の商品の商品番号と販売価格のデータが欲しい。

2020/5/14

データベース (©横田)

128

結合演算の例(1)

□ 異なる関係で同じ属性名→関係名をプレフィックスとして識別

```
SELECT 製造元. 商品番号、製造元. 商品名、製造元. 製造元、
      秋葉原支店. 販売価格、秋葉原支店. 在庫
FROM 製造元、秋葉原支店
WHERE 製造元. 商品番号=秋葉原支店. 商品番号;
```

製造元	商品番号	商品名	製造元
	P.X	P.	P.

秋葉原支店	商品番号	販売価格	在庫
	X	P.	P.

2020/5/14

データベース (©横田)

129

結合演算の例(2)

- Fepsonが製造する商品の分類は？

```
SELECT 商品分類, 商品分類, 商品分類, 商品名
FROM 商品分類, 製造元
WHERE 商品分類, 商品名 = 製造元, 商品名
AND 製造元, 製造元 = "Fepson";
```

製造元	商品番号	商品名	製造元
		X	Fepson

商品分類	商品分類	商品名
	P.	P.X

2020/5/14

データベース (©横田)

130

結合演算の例(3)

- 同じ問い合わせの別の記述方法
 - 入れ子問い合わせ

```
SELECT 商品分類, 商品名
FROM 商品分類
WHERE 商品名 IN (
    SELECT 商品名
    FROM 製造元
    WHERE 製造元 = "Fepson");
```

製造元	商品番号	商品名	製造元
		X	Fepson

商品分類	商品分類	商品名
	P.	P.X

QBEは変わらず

2020/5/14

データベース (©横田)

131

3重の入れ子の例(3)

- 秋葉原支店に在庫が10以上ある商品の商品分類は？

```
SELECT 商品分類, 商品名
FROM 商品分類
WHERE 商品名 IN (
    SELECT 商品名
    FROM 製造元
    WHERE 商品番号 IN (
        SELECT 商品番号
        FROM 秋葉原支店
        WHERE 在庫 ≥ 10 ));
```

秋葉原支店	商品番号	販売価格	在庫
	Y		≥ 10

製造元	商品番号	商品名	製造元
	Y	X	

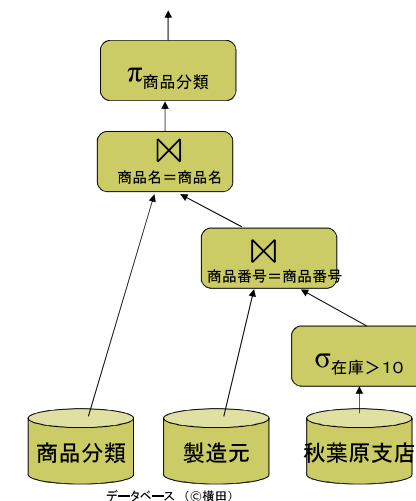
商品分類	商品分類	商品名
	P.	P.X

2020/5/14

データベース (©横田)

132

3重の入れ子の例の問い合わせ木



2020/5/14

データベース (©横田)

133

ちょっと考えてみよう(4-4)

- 次の問い合わせをSQLで書いてみましょう。
 - 商品分類がComputerに属するGELLの商品で、秋葉原支店か日本橋支店で販売価格が 100,000円以下で売られている商品の商品名と商品番号を知りたい。

便利な検索機能

- 集約演算
 - カウント、総和、平均、最大値、最小値
- 集約演算に関する便利な機能
 - Group-by
 - Having
- 出力に関する機能
 - Order-by

集約演算の例

```
SELECT COUNT(*)
FROM 製造元
WHERE 製造元="Hanon";
```

製造元	商品番号	商品名	製造元
P.CNT.ALL			Hanon

```
SELECT COUNT(DISTINCT 商品名)
FROM 製造元
WHERE 製造元="Hanon";
```

製造元	商品番号	商品名	製造元
		P.CNT.	Hanon

```
SELECT SUM(在庫)
FROM 秋葉原支店
WHERE 販売価格 ≤ 100,000;
```

秋葉原支店	商品番号	販売価格	在庫
		≤ 100,000	P.SUM.ALL

AVG()、MAX()、MIN()

AVG.ALL、MAX.ALL、MIN.ALL

Group-By演算の例

- 製造元毎の平均価格を求めたい

```
SELECT 製造元, 製造元 AS 製造元,
      AVG(秋葉原支店. 販売価格) AS 平均価格
FROM 製造元, 秋葉原支店
WHERE 製造元. 商品番号 = 秋葉原支店. 商品番号
GROUP BY 製造元, 製造元;
```

結果

製造元	商品番号	商品名	製造元
	X		P.G.

秋葉原支店	商品番号	販売価格	在庫
	X	P.AVG.ALL	

製造元	平均価格
GELL	120,000
Fepson	70,000
Hanon	30,000
Tenao	40,000

Having句 (Group-Byと共に)

```
SELECT 製造元, 製造元 AS 製造元
FROM 製造元, 秋葉原支店
WHERE 製造元, 商品番号 = 秋葉原支店, 商品番号
GROUP BY 製造元, 製造元
HAVING AVG(秋葉原支店, 販売価格) ≥ 50,000;
```

製造元	商品番号	商品名	製造元	CONDITION
	X		P.G.	AVG.ALL.Y ≥ 50,000

秋葉原支店	商品番号	販売価格	在庫
	X	AVG.ALL.Y	

結果

製造元
GELL
Fepson

2020/5/14

データベース (©横田)

138

Order-By 句

昇順

```
SELECT *
FROM 秋葉原支店
WHERE 在庫 > 10
ORDER BY 販売価格 ASC;
```

秋葉原支店	商品番号	販売価格	在庫
P.		AO.	>10

降順

```
SELECT *
FROM 秋葉原支店
WHERE 在庫 > 10
ORDER BY 販売価格 DESC;
```

秋葉原支店	商品番号	販売価格	在庫
P.		DO.	>10

2020/5/14

データベース (©横田)

139

ちょっと考えてみよう(4-5)

1. 秋葉原支店において在庫数が 5 以下である商品の商品分類ごとの最低販売価格を示す問い合わせをSQLで書いてみよう。
2. 秋葉原支店において在庫数が 5 以下である商品で最低販売価格が50,000 円以上の商品分類を示す問い合わせをSQLで書いてみよう。

2020/5/14

データベース (©横田)

140