

データモデル

Hiroshima Institute of Technology

1

授業計画

- 第1回 ガイダンス・データベースの基本概念
- 第2回 データモデル
- 第3回 関係代数
- 第4回 データベース設計
- 第5回 リレーションの正規化
- 第6回 中間まとめ
- 第7回 関係データベース言語(SQL1)
- 第8回 関係データベース言語(SQL2)
- 第9回 計算機実習
- 第10回 データの検索機構 MySQL実習
- 第11回 トランザクション管理 MySQL実習
- 第12回 障害回復 MySQL実習
- 第13回 分散データベース MySQL実習
- 第14回 期末まとめ
- 第15回 応用技術と将来動向 MySQL実習

2. データモデル

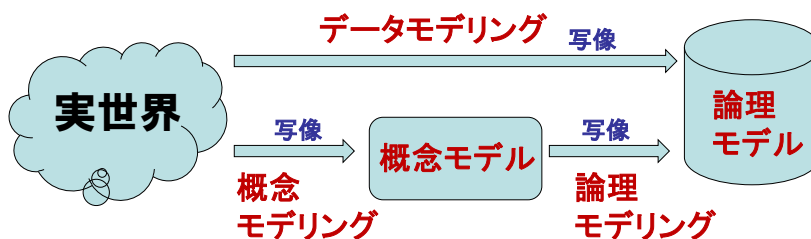
講義内容

- データモデルとは
- 概念モデル
- 論理モデル（関係モデル・階層モデル・ネットワークモデル）
- 関係モデル
- ANSI/SPARC 3層スキーマ構造
- 概念設計としてのERモデル

2. データモデル

1. データモデル

- データモデル：
実世界の表現方法や実世界を記述するための記号系
- データモデリング：
実世界のデータをデータモデルに写像すること
- 概念モデル：
特定のDBに依存しないデータモデル



2. データモデル

1. データモデル

■ データモデルの種類

- ① リレーショナルデータモデル (relational data model)
- ② ネットワークデータモデル (network data model)
- ③ 階層データモデル (hierarchical data model)

2. データモデル

1. データモデル

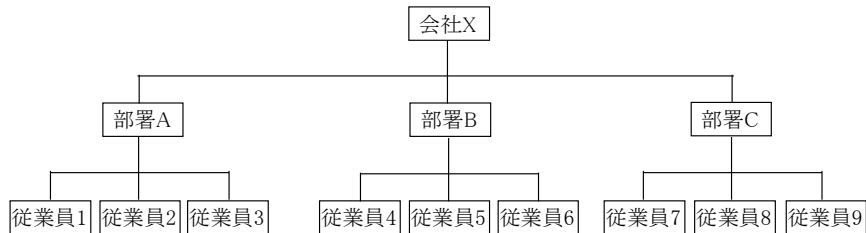
■ 階層データモデル

- 木構造のデータモデル, 実体を親子関係で表現
- 実体をレコードで, レコード間の関係をポインターで表現
- 上位: 親レコード 下位: 子レコード
- 商用DB: 1968年 IBM
IMS (information management system)

2. データモデル

1. データモデル

■ 階層データモデル



2. データモデル

1. データモデル

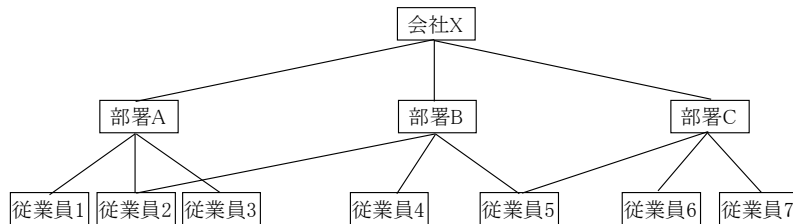
■ ネットワークデータモデル

- **網構造**のデータモデル
- **親となる実体を複数持てる**
各実体の関係をより自然に表現
- 商用DB 1971年 米国
CODASYL(conference on data systems languages)

2. データモデル

1. データモデル

■ ネットワークデータモデル



2. データモデル

1. データモデル

■ 関係データモデル

- 数学理論に基づくデータモデル
- 多くのDBMSで採用
- E.F.Codd (1970年 IBM)が提案
- データ構造は2次元の表
- 構造上の前提条件がなく、自由にデータベースの設計が可能

2. データモデル

1. データモデル

■ 関係データモデル

- **リレーション**(関係)と呼ぶ**列**と**行**からなる**表構造**
- データを**n項関係**で表現
- 関係はn個の**定義域の直積の部分集合**で定義
- リレーションの列:**アトリビュート**(属性), 行:**タプル**
- 関係→**関係名**, 属性→**属性名**を持つ
- 関係の属性数 = 関係の**次数**

Hiroshima Institute of Technology

11

2. データモデル

1. データモデル

■ 関係データモデル

- 関係データベース管理システム(**RDBMS**):
Oracle Database, DB2, Microsoft SQL Server, MySQL, PostgreSQL, FileMaker, H2 Databaseなど

アトリビュート (属性)

部署ID	従業員	年数	会社名	
部署A	従業員1	24	会社X	
部署A	従業員2	26	会社X	
部署A	従業員3	34	会社X	
部署B	従業員2	44	会社X	
部署B	従業員4	54	会社X	

タプル

Hiroshima Institute of Technology

12

2. データモデル

1. データモデル

■ オブジェクト指向データモデル

➤ オブジェクト指向の考え方を採用

➤ 1990年代, オブジェクト関係データベース管理システム
(ORDBMS: object-RDBMS)の開発が進展

➤ ORDBMS:

Informix, PostgreSQL, IBM DB2, OracleDatabase

など

2. データモデル

2. ANSI/SPARCの3層スキーマ構造

■ ANSI/SPARCの3層スキーマ

■ スキーマ: データの定義の集合, DBの定義情報のこと

■ ANSI/SPARCによる3層スキーマ:

DBMSのインターフェースを整理したもの

(American National Standards Institute /
Standards Planning And Requirements Committee)

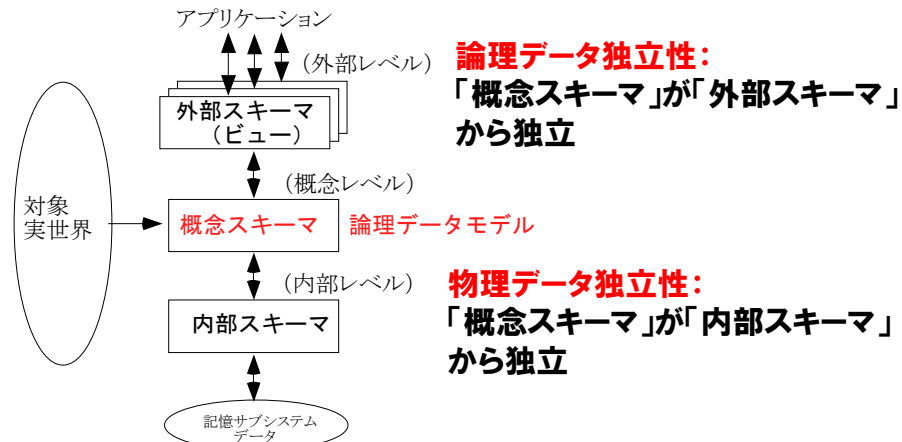
➤ 外部スキーマ, 概念スキーマ, 内部スキーマから構成

➤ 各層でデータの独立性を保持

2. データモデル

2. ANSI/SPARCの3層スキーマ構造

■ 3層スキーマ構造の独立性



Hiroshima Institute of Technology

15

2. データモデル

2. ANSI/SPARCの3層スキーマ構造

■ 概念スキーマ (conceptual schema)

- 実世界のデータをデータモデルによって**抽象化**, 抽象化した**概念**と**概念間の関係**を定義し記述したもの
- 「外部スキーマ」と「内部スキーマ」の中間に位置し, 各スキーマから独立に定義される
- 一つのDBに一つの「概念スキーマ」が対応
- 概念スキーマは, DBMSの**論理データモデル**と同一

Hiroshima Institute of Technology

16

2. データモデル

2. ANSI/SPARCの3層スキーマ構造

■ 内部スキーマ(internal schema)

- 「概念スキーマ」の内側に位置, ディスク装置などに
データを物理的に実装するためのスキーマ
- 「概念スキーマ」と区別することで, DBの物理的データ
独立性を実現

2. データモデル

2. ANSI/SPARCの3層スキーマ構造

■ 外部スキーマ(external schema)

- 「概念スキーマ」の外側に位置
- ユーザ(アプリケーション)に対して, **データを見せる**ためのスキーマ
- RDBMSの**ビュー**(ビュー表)が外部スキーマに対応
- 一つのDBに対して, **複数の外部スキーマ**が存在
- DBの見せたくない部分を**隠ぺい**することが可能

2. データモデル

3. 概念設計としてのE Rモデル

■ 概念データモデルの図式表現

■ E-Rモデル(実体関連モデル(entity relationship model))

- 実世界を**実体**, **関連**, **属性**という要素を用いて概念化
- 1976年に**P. P. Chen**が提案, 記述が容易→広く利用
- **エンティティ(実体)**: 管理対象となるもの
 - ・ 物理的なもの(商品や顧客)
 - ・ **抽象的な概念**(発注や検査)

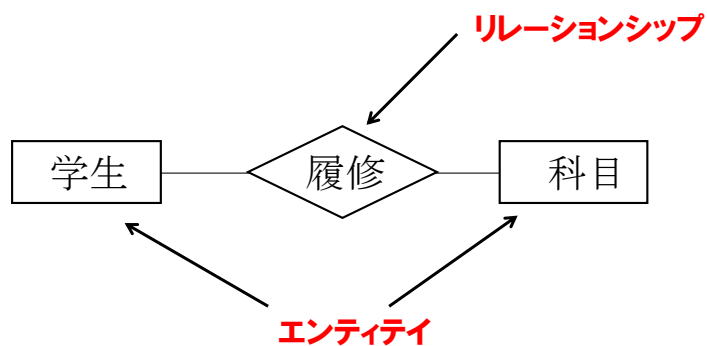
Hiroshima Institute of Technology

19

2. データモデル

3. 概念設計としてのE Rモデル

■ P. P. ChenのERDのオリジナルの表記法



Hiroshima Institute of Technology

20

2. データモデル

3. 概念設計としてのERモデル

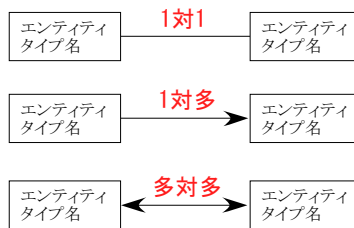
■ 情報処理技術者試験の表記法

(1) エンティティとリレーションシップの表記法

① **エンティティ**は、**長方形**で表し、長方形の中にエンティティ名を記入する。

② **リレーションシップ**は、エンティティ間に引かれた**線**で表す。

- ・“1対1”のリレーションシップを表す線は、矢をつけない。
- ・“1対多”のリレーションシップを表す線は、“多”側の端に矢を付ける。
- ・“多対多”のリレーションシップを表す線は、両端に矢を付ける。



エンティティとリレーションシップ
の表記ルール

hiroshima institute of Technology

21

2. データモデル

3. 概念設計としてのERモデル

(2) リレーションシップを表す線で結ばれたエンティティ間の表記法

- ① 一方のエンティティのインスタンスから見て、他方のエンティティに対応するインスタンスが存在しないことがある場合、線の対応先に“○”を付ける。
- ② 一方のエンティティのインスタンスから見て、他方のエンティティに対応するインスタンスが必ず存在する場合は、線の対応先に“●”を付ける。



(a) 双方のインスタンスが存在しないことがある場合



(b) 双方のインスタンスが必ず存在する場合



(c) 左のインスタンスが存在しないことがあり、右のインスタンスが必ず存在する

リレーションシップの表記ルール

Hiroshima Institute of Technology

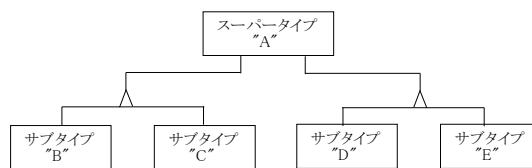
22

2. データモデル

3. 概念設計としてのERモデル

(3) スーパータイプとサブタイプ間のリレーションシップの表記法

- ① サブタイプの切り口の単位に“△”を記入し、スーパータイプから“△”に1本の線を引く。
- ② 一つのスーパータイプにサブタイプの切り口が複数ある場合、切り口の単位毎に“△”を記入し、スーパータイプからそれぞれの“△”に別の線を引く。
- ③ 切り口を示す“△”から、その切り口で分類されるサブタイプのそれぞれに線を引く。



スーパータイプとサブタイプの間のリレーションシップの表記ルール

Hiroshima Institute of Technology

23

2. データモデル

まとめ

- データモデルとは
- 概念モデル
- 論理モデル（関係モデル・階層モデル・ネットワークモデル）
- 関係モデル
- ANSI/SPARC 3層スキーマ構造
- 概念設計としてのERモデル

Hiroshima Institute of Technology

24