

ERモデル

データベース論 I 第5回

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

作成者 末次文雄 ©

目次

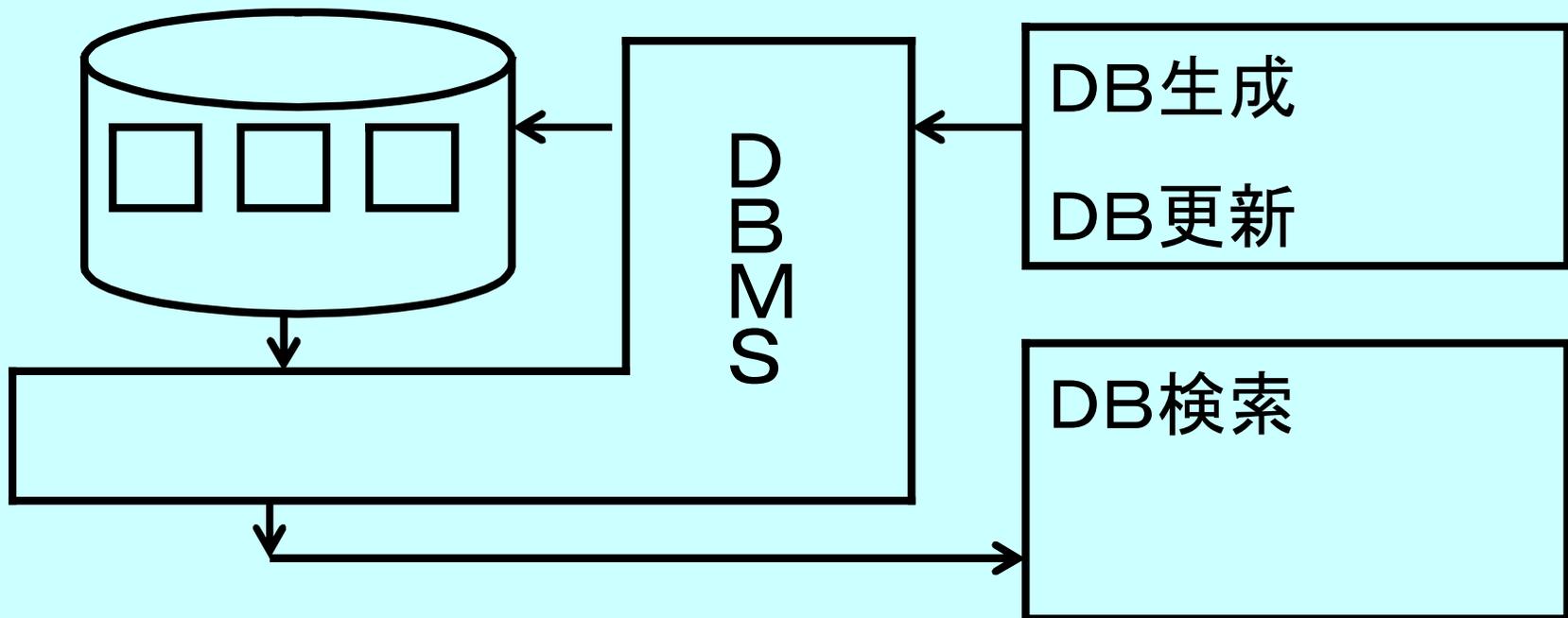
1. データ正規化のメリット(復習)
2. ERモデリングの基本
3. ERモデルにおける関連の概念
4. SEに必要な要件(参考)
5. MySQLダウンロード案内
6. レポート課題
7. 参考書ほか

1. データ正規化のメリット (復習)

- 1. 1 データベース化のネライ
- 1. 2 正規化データの活用

1.1 データベース化のネライ

- 要は、データベースは、個々の情報を整理して保管し、
検索機能を提供することにより、
人がうまく情報を利用する手助けを行う。



1. 2 正規化データの活用

- ①表(テーブル)では、**キー**が基本
- ・データ操作の基本
 - ・データの論理構造の基本

(ただし、INDEXはキー以外のデータ項目(列)にも設定可能)

②表(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、代替キー)

 - 行のキー候補(社員番号と社会保険番号等)

- ・外部キー (External Key、参照キー)

 - 他の表の主キーになっているもの

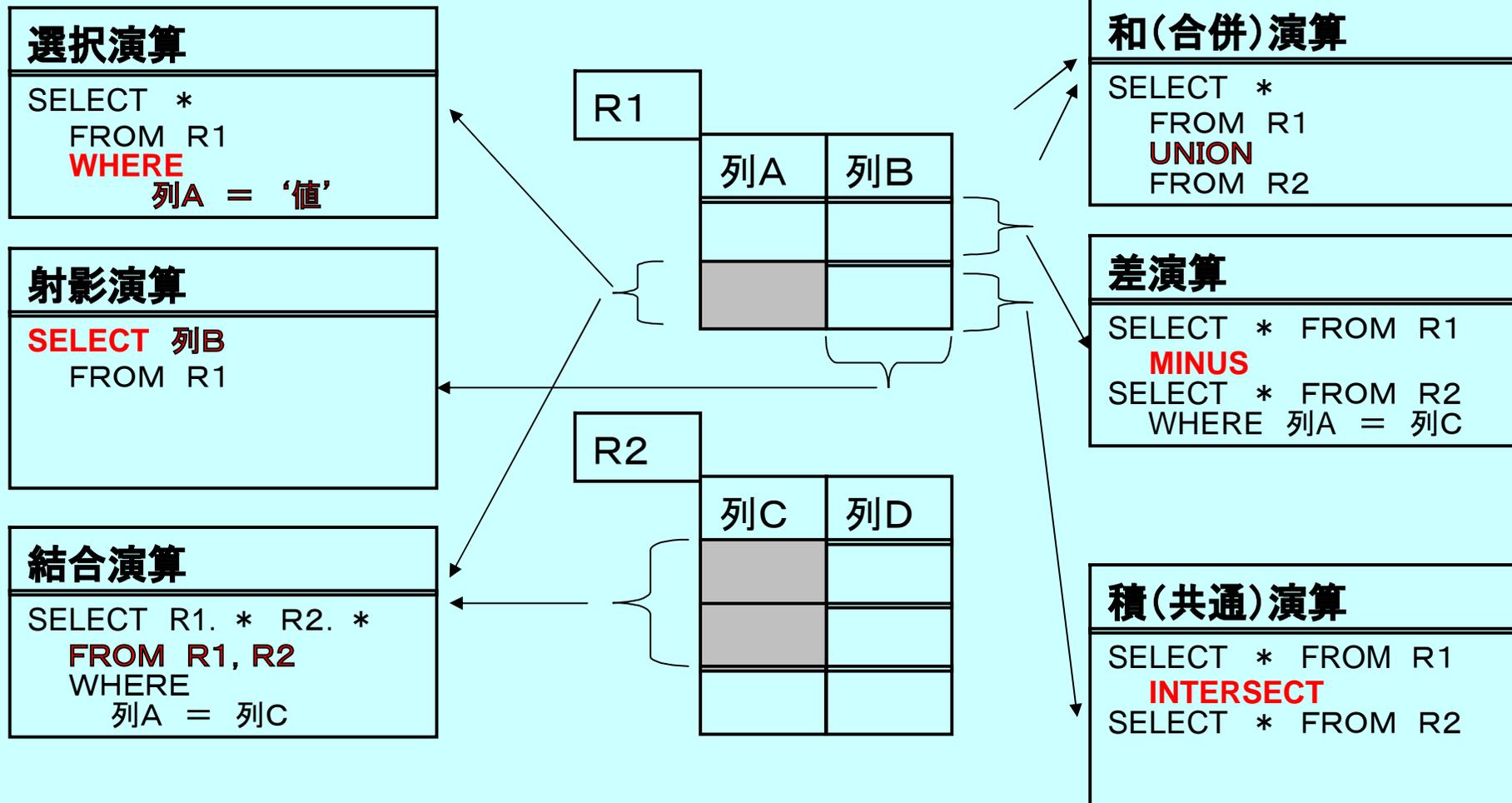
- ・連結キー (Concatinated Key)

 - 複数の列からなる複合された組合せキー

いずれも、ERモデルで重要な情報となる。

③表(テーブル)の検索操作

関係の集合演算がSQLで可能である。



(直積のケースは省略)

④ データ検索事例

社員スキル管理表

正規化前

社員 = 社員番号 + 社員氏名 + 組織コード

正規化後

組織 = 組織コード + 組織名 + 管理者名

保有スキル = 社員番号 + スキルコード + 経験年数 + スキルレベル

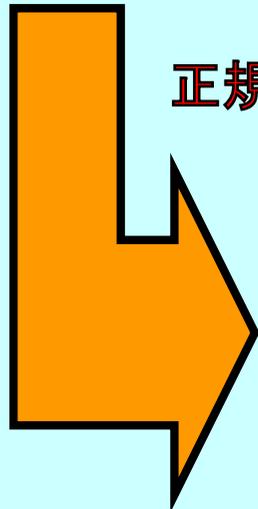
スキル = スキルコード + スキル名

正規化後では、データの重複が排除できた。

④(続き) データ検索事例

社員 = 社員番号 + 社員氏名 + 組織コード	R1
組織 = 組織コード + 組織名 + 管理者名	R2
保有スキル = 社員番号 + スキルコード + 経験年数 + スキルレベル	R3
スキル = スキルコード + スキル名	R4

正規化結果のDBを使用して、誰でも容易に、いろいろな情報を作成できる。



社員スキル管理表

組織別社員スキル管理表

R2 → 組織表

スキル別社員スキル管理表

R3 → スキル一覧表

特定スキル保有者表



2. ERモデリングの基本

2. 1 ERモデルの由来

2. 2 ERモデルの位置付け

2. 3 ERモデルの考え方

2. 4 ERモデルの基本概念

2. 5 ER図

2. 1 ERモデルの由来

1976年にアメリカのPeter Pin-Shan Chenが、発案
「Entity-Relationship Model」

- ・対象領域において発生した事象を、データベースとして作成、記録するために、
データ構造を分析して、記述する手法。
- ・分析対象のデータ構造を、エンティティ(実体、Entity)とエンティティ間の関連(Relationship)としてとらえる。

2.2 ERモデルの位置付け

① ER手法の二つの活用法

- **トップダウン分析**

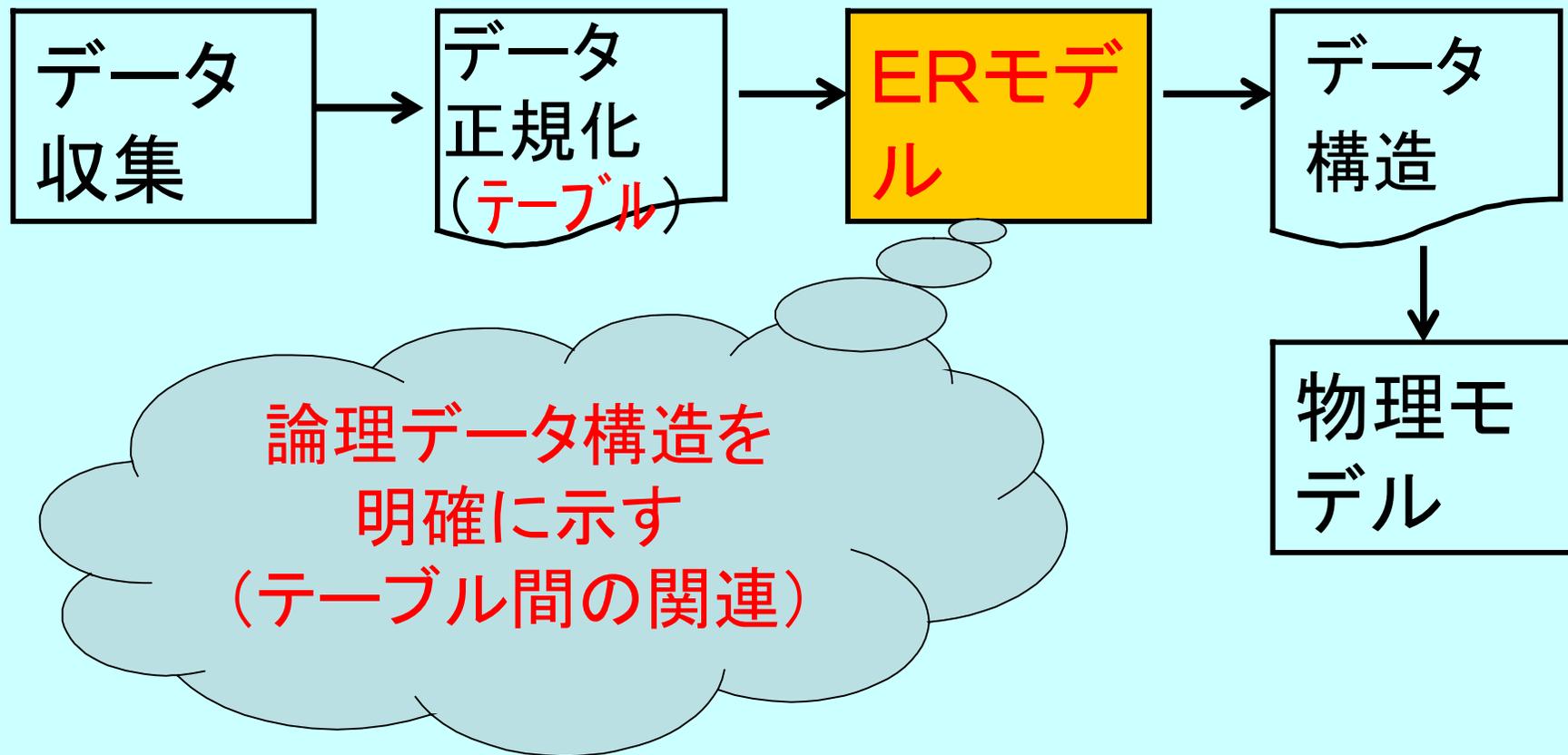
- **概念データベース設計**とも言う
- 概念スキーマ(論理データベース)設計の前に、データベースのラフな設計を実施し、その時にER手法を使う。
- **全くの新規業務の**データベース化に適している

- **ボトムアップ分析**

- リレーショナルモデルでのデータ**正規化の後**で、論理データベース構造の検証用、安定構造設計用にER手法を使う。
- **現行業務・システム**のデータベース化に適す。

② ERモデリングの位置付け

データ正規化と物理データモデル作成の中間にあるプロセスである。



2. 3 ERモデルの考え方

- Peter P. Chenの発想の原点
 - データが実世界で持つ、**意味上の関係**を表現
- 「世の中に存在するあらゆるものは、
具体的であれ抽象的であれ、
ENTITY(実体)とその**RELATIONSHIP(関連)**という二つの概念で表すことが出来る。」
- オブジェクト指向につながる(意味論的)
- DOA手法のキー・ツールでもある。
Data Oriented Architecture(データ中心設計)

2. 4 ERモデルの基本概念

ERモデルの三つの構成概念

①エンティティ(Entity、実体、事象)

- ・企業、組織にとって関心があり、**継続的に管理の対象になるもの**である。
- ・個々の実体をエンティティまたは、**エンティティ・オカレンス**、エンティティ・インスタンス
- ・実体の集合を**エンティティ・タイプ** or セットという。
(通常、このエンティティ・タイプを**エンティティ**と略称)

②属性(アトリビュート)

- ・エンティティの**性質(プロパティ)**を示すもので、
各エンティティは、複数の属性を持つ。

2.4 (続き) ERモデルの基本概念

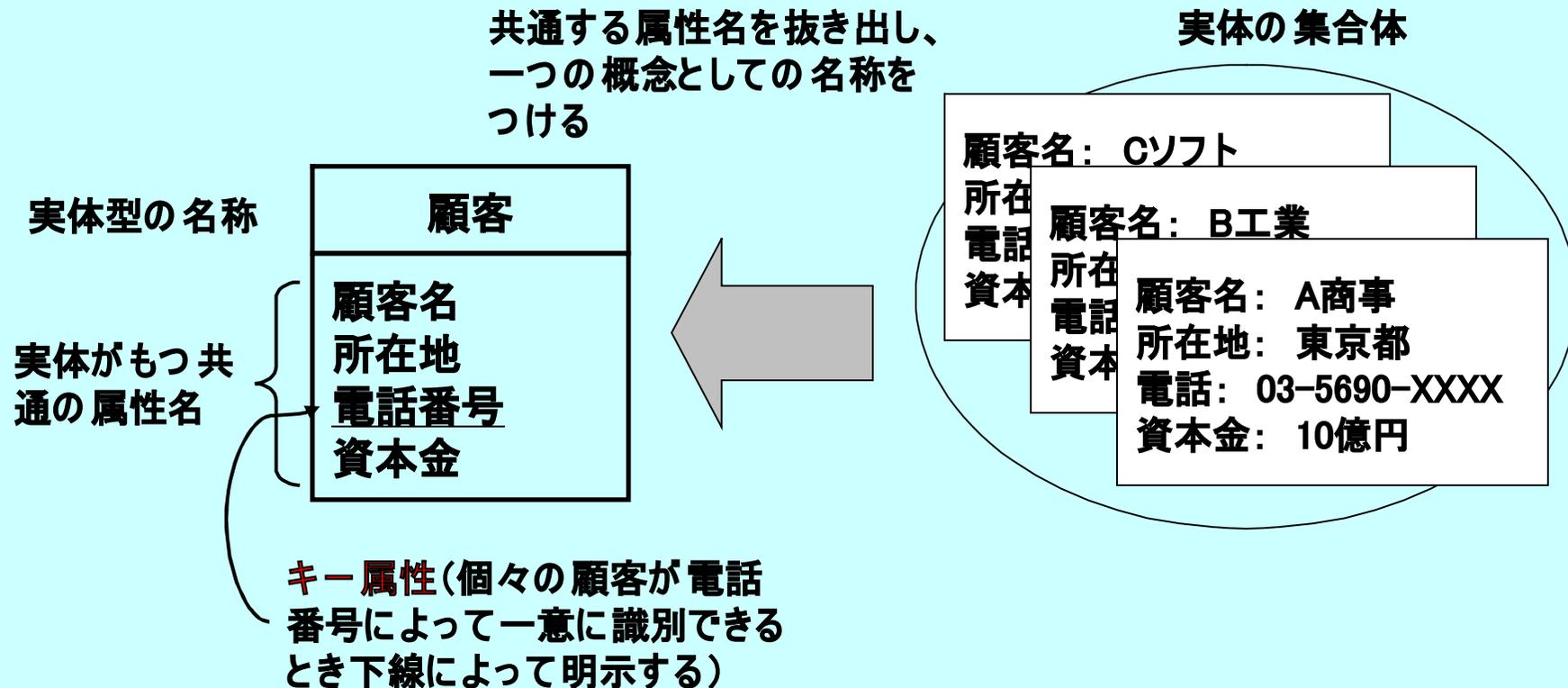
③ 関連 (Relationship)

- エンティティ間の関係性のことで、
- オカレンスの発生に関して、種々の制約あり。
 - 基数制約 (カーディナリティ制約)
 - 存在依存制約 (Existence Dependency)
- いろいろな関連が存在する。
- 業務上の意味を十分検討して、関連を求める。

ER(実体関連)モデルによる表現

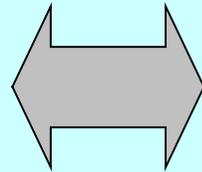
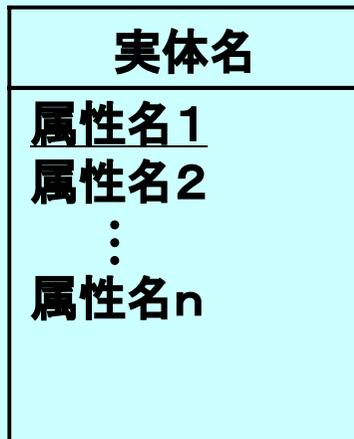
(3) Entity Type (実体型)

実体の集合体を表現する。実体の集合体は、一つの概念を表わしたもので、名称と個々の実体が共通してもつ種々の属性(attribute)を記述する。



実体の表現

《ERモデル》



《関係モデル》

(関係表名)

<u>属性名1</u>	属性名2	...	属性名n

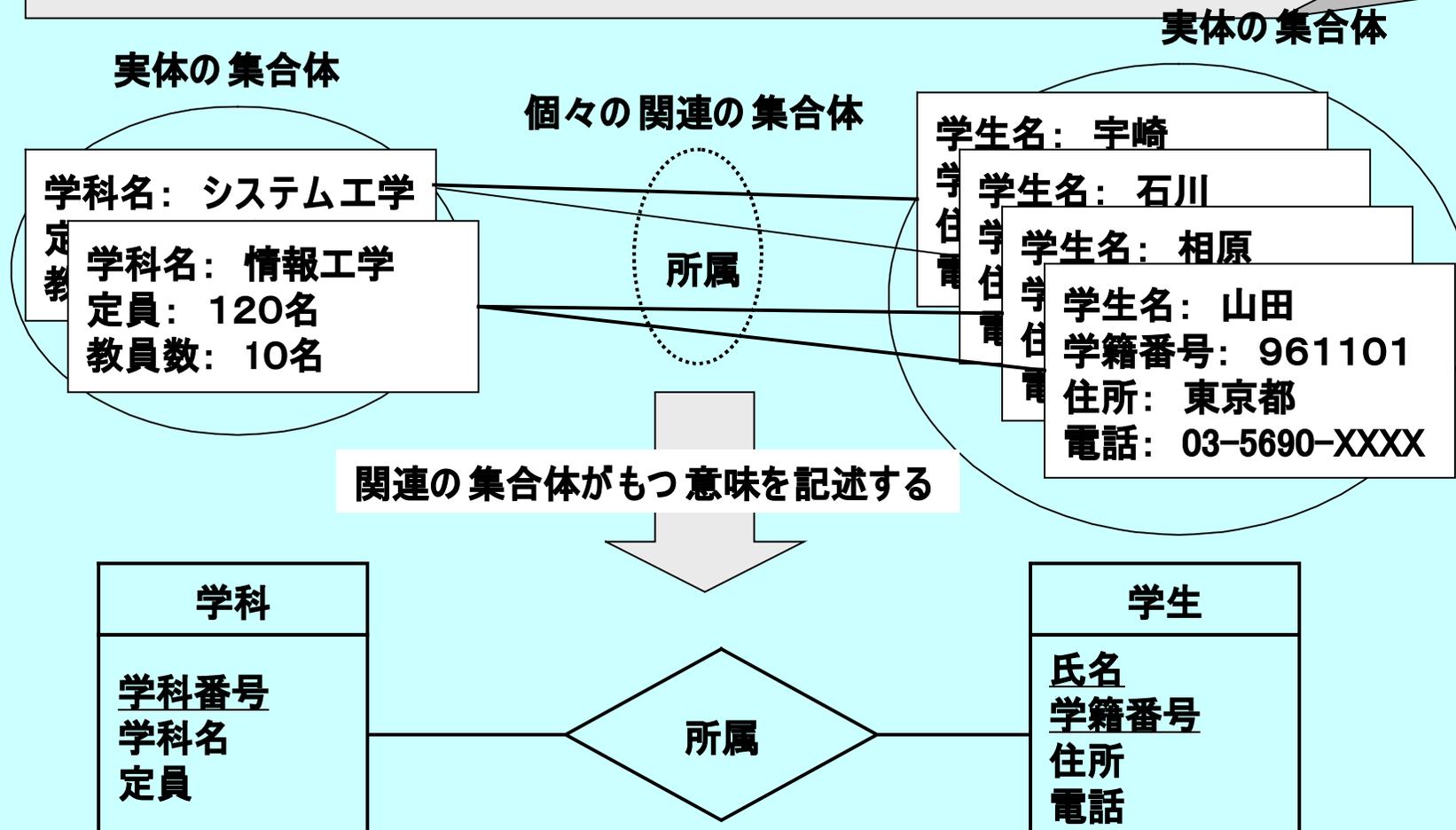
関数従属



ER(実体関連)モデルによる表現

(3) Relationship Type (関連型)

結び付けられた実体間の関連の集合体を表わす。

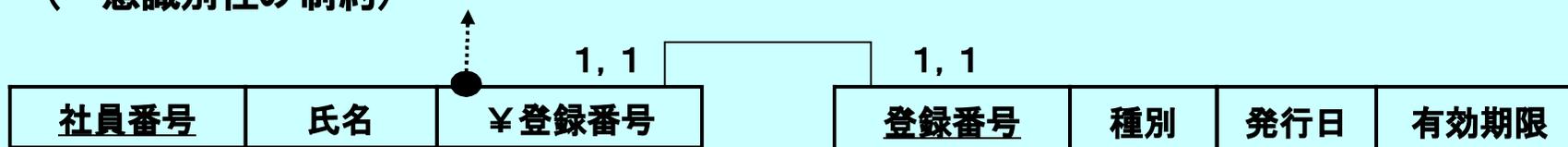


関連の表現

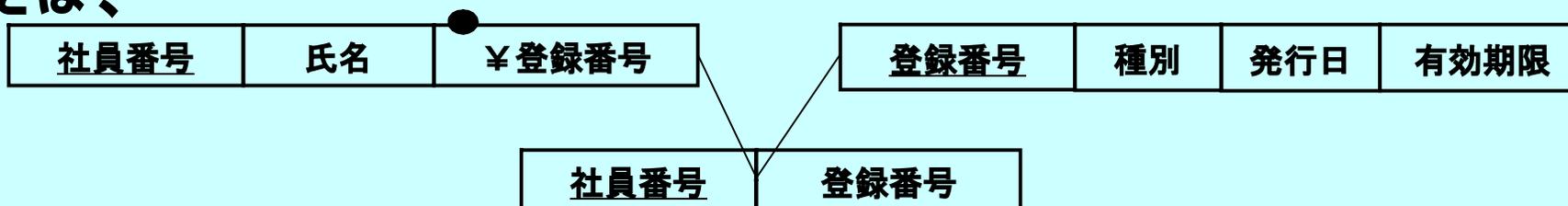
《例1》 1対1関連



※登録番号は一意的に識別できる
(一意識別性の制約)



または、



2.5 ER図(実体-関連ダイアグラム)

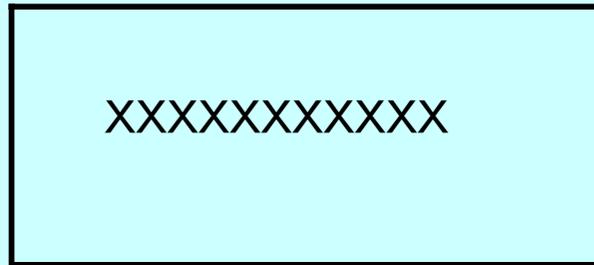
①特徴

- データ正規化の結果を元にして、
- データ本来の意味とデータ間の関係を把握し、
- データ固有の構造と特性を把握する。

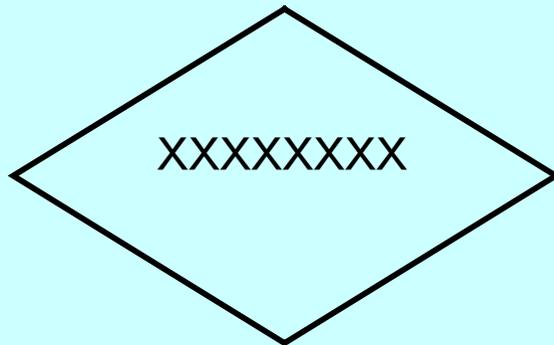
②メリット

- 管理すべきデータを構造的に理解できる
- 独立性が高く冗長性が無いデータを保有
- 高品質で安定したシステムおよびDBの構築が可能となる。

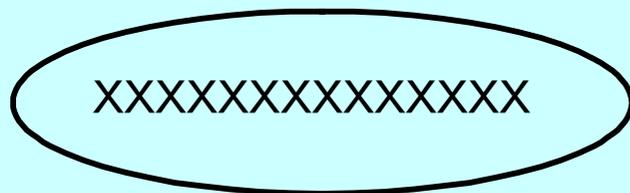
③ 作図記号



----エンティティ (実体)



----関連



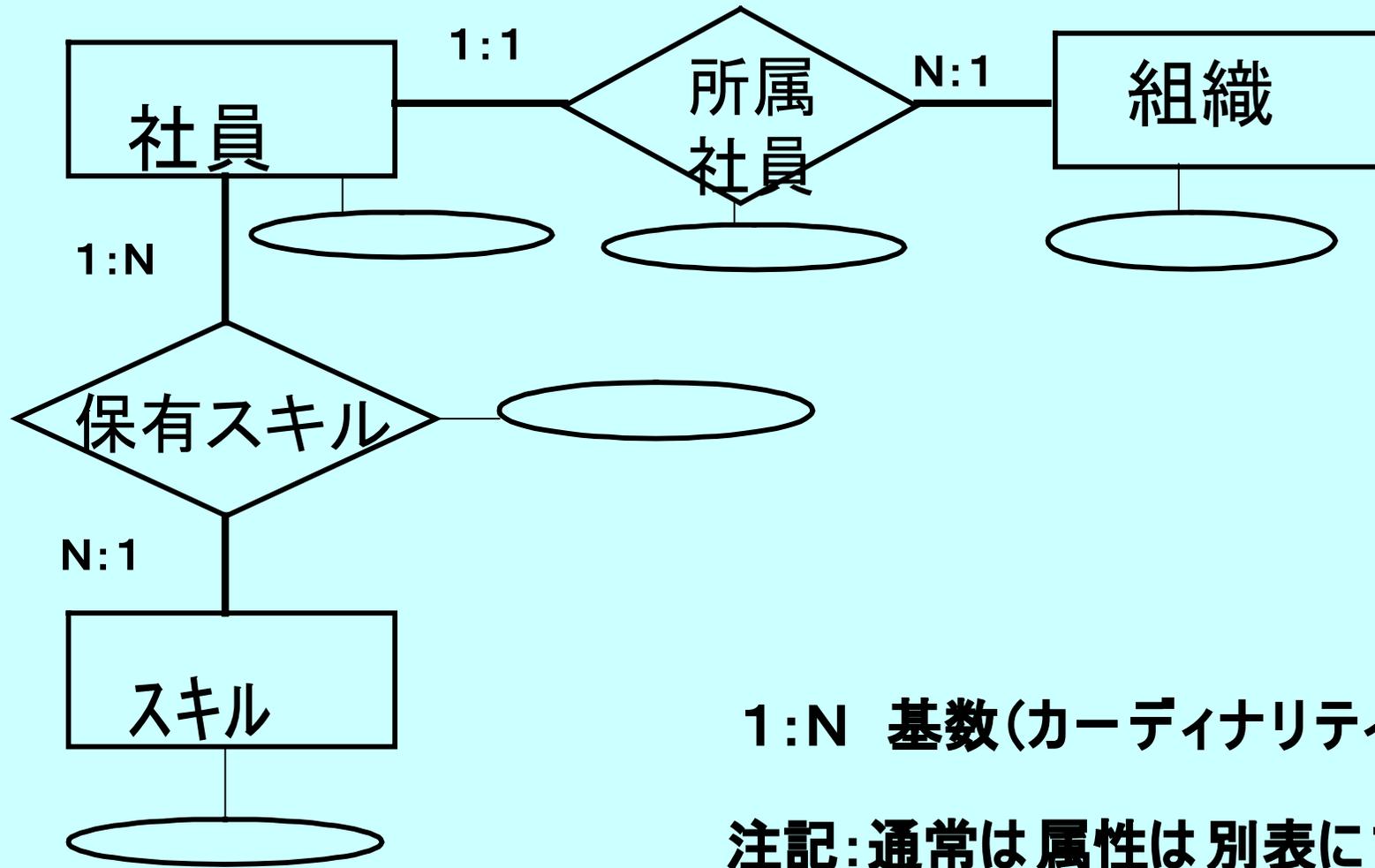
----属性(データ項目)

④ ER図の作成手順

1. システム化対象の中で、データ正規化結果を元に、**エンティティを決める**。
2. エンティティ同士の**関連 (Relationship) を見出す**。
3. その**関連の対応関係**を見出す。
(1:1 1:M M:M)
4. 属性(データ項目)を漏れなくまとめる。

これで、論理データ構造図が完成。

(補足)ER図の事例



1:N 基数(カーディナリティ)

注記:通常は属性は別表にする

⑤ ERモデル作成時の留意点

- データ正規化済みのテーブルを良く吟味。
 - データ項目の意味を明確にする。
 - 主キーはエンティティの候補にする。
- 関連は、外部キー、連結キーに着目して求める。
 - 他のテーブルの主キーを外部キーに持てば、従属エンティティ(子)となりうる。
 - 連結キーを構成するキーは、単独でエンティティとなりうる。
- システム化対象領域での、データ漏れを発見し、エンティティ、関連を追加する。
- 業務上の常識からみて、矛盾するエンティティ、関連を見直す。

(補足)エンティティの求め方

エンティティになりうるもの

- 人 : 社員、顧客、株主、学生、先生…
- 組織 : 自社組織、特約店…
- 場所 : 本店、支店、販売地区、工場、倉庫、
店舗、配送拠点…
- 物 : 商品、製品、部品、材料、設備、
機器、金、伝票、帳簿…
- 事象 : 受注、発注、在庫、入金、売掛…
- 概念 : 業務、スキル、目標、法規…

- ・要するに、企業、組織にとって関心があり、**継続的に管理の対象になるもの**である。
(=オブジェクト指向でいうクラスに近似)

3. ERモデルにおける関連 の概念

- 3. 1 関連の制約
- 3. 2 関連の基数
- 3. 3 特殊な関連

3.1 関連の制約

① 基数制約

- ・エンティティ間には、オカレンスの発生について、以下の制約がある。

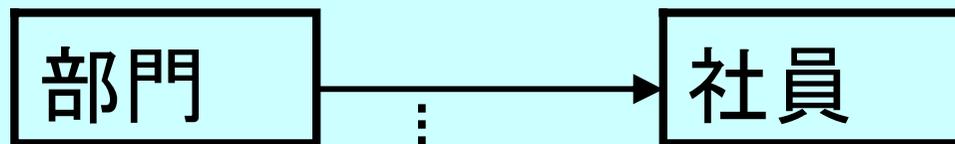


- ・部門、社員の関係には、必ず1:Nの対応関係がある。
- ・部門には、必ず社員が所属している。
- ・社員は、必ずいずれかの部門に所属する。

3. 1 (続き) 関連の制約

② 存在依存制約

- ・一方のエンティティのオカレンスの存在が、他方のエンティティのオカレンスの存在に依存する関係がある場合には、存在依存制約の関係にある。
- ・**識別子従属関連**とも言う。

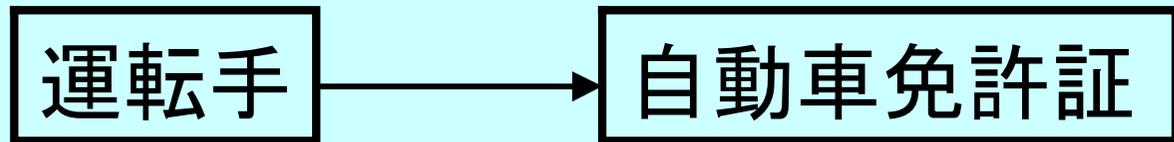


- ・社員は、部門の存在を必要とする。
- ・部門を削除すれば、社員も削除する。
- ・社員を追加する場合、部門が無ければ部門も追加する。

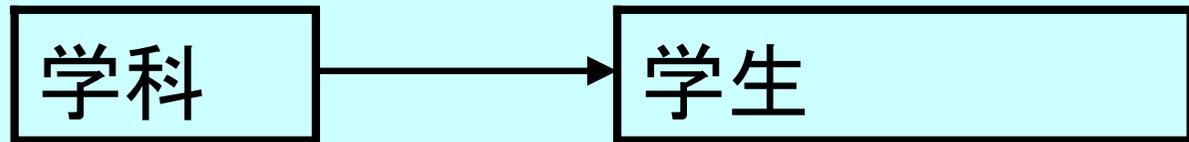
3.2 関連の基数

- エンティティ間の対応関係には、以下の種類がある。

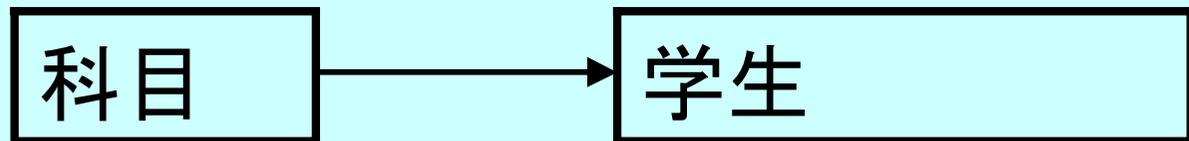
1:1



1:N



N:M

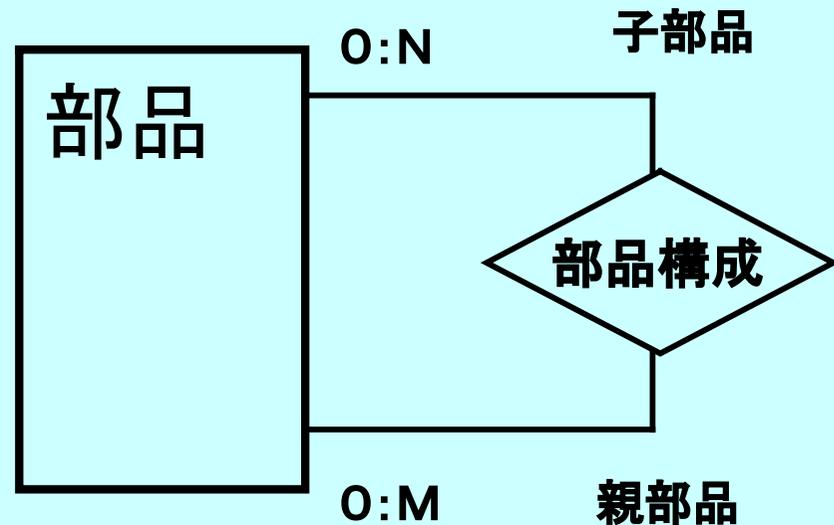
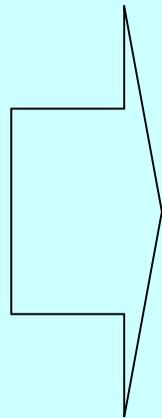
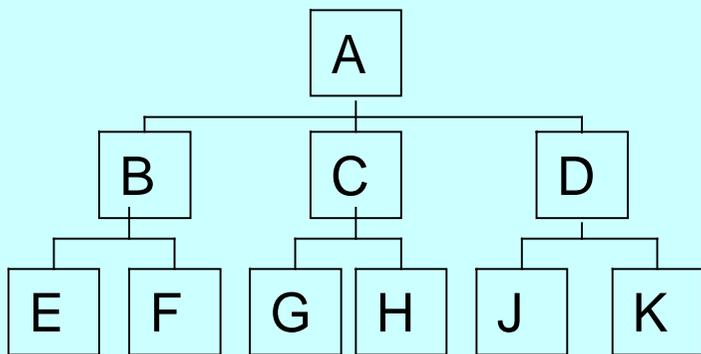


3.3 特殊な関連

① 再帰的関連

同じエンティティ同士でありながら、ツリー構造(木構造)、上下関係、親子関係にあるもの。

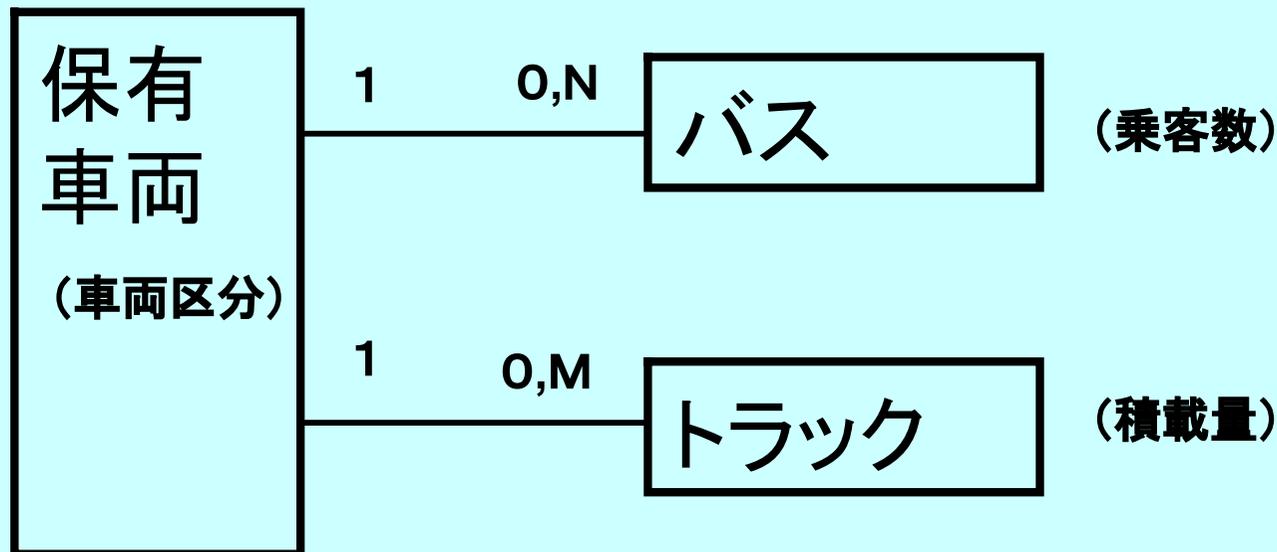
例: 上司・部下の関係
部品の構成関係



3.3 (続き) 特殊な関連

② 選択的関連 (サブ・エンティティ)

あるデータ項目の有無や値によって、いずれかの選択値が存在する時の関連を示す。

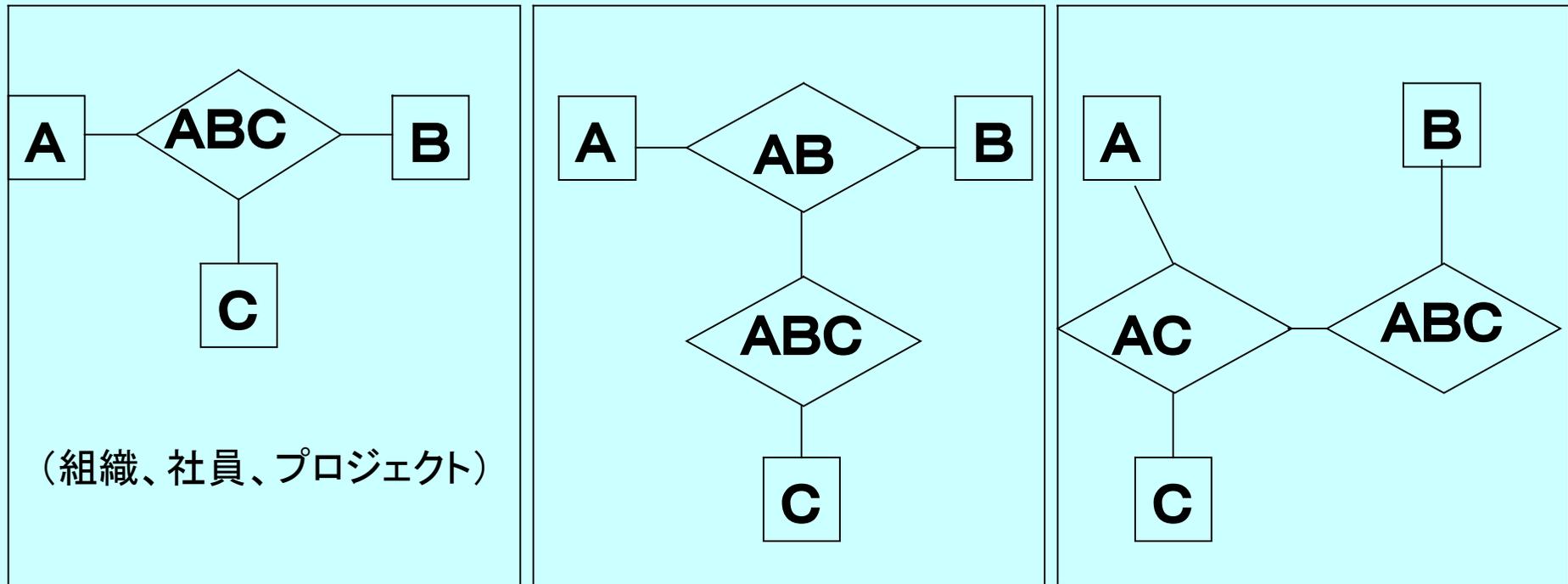


3.3 (続き) 特殊な関連

③ 複数関連 (3項関連)

3つ以上のエンティティが互いに関連しているケース。

(3つ以上の連結キーを持つ関連が存在する)



4. SEに必要な要件（参考）

① 技術・知識

- －ITスキル（情報処理技術、開発技術、DB技術）
- －管理技法（分析技術、コミュニケーション技法）
- －業務知識
 - ・主活動（開発、購入、製造、物流、販売）
 - ・支援業務（経営、財務、人事、システム）

4. (続き) SEに必要な要件

② SEの心構え

- 好奇心 → もっと知る → 対象業務を好きになる
- 忍耐力、持久力
- 柔軟な考え方ができる
- スタートしなければゴール出来ない

③ 仕事の進め方

- 何事も計画を立ててから進める
- 書き出してみる(リスト、関連図)
- 自分の考えを話し、人からアイデアをもらう
- 5W3H(What, Why, Who, When, Where, How to, How much, How many)

④ 人材トレンド

- プロジェクト・リーダー 経験者など
- 業務系システムの構築・運用経験者
- データベースのスキルを持つエンジニア
“力のある”エンジニアへの需要の伸びは顕著

5. MySQLのダウンロード案内

- Windows環境で利用できる唯一のフリーDBMS
- オープンなDBMS(OSのLinuxのようなもの)
- 参考URL

(SQLダウンロード)

<http://wings.raindrop.jp/openoffice/o3u2gj/database.html>

<http://homepage1.nifty.com/gak/webdb/mysql.html>

(Active Perlダウンロード)

<http://www.activestate.com/Products/Download/Download.pl?x?id=ActivePerl>

(かねやんのSQLツール) http://www.mmdb.net/m_kaneko/

(MySQL日本ユーザー会) <http://www.mysql.gr.jp/>

(SQL講座) <http://www.rfs.jp/sitebuilder/sql/>

6. レポート課題

- ① ERモデリングの基本概念を説明して下さい。
- ② MySQLのダウンロード経過、結果の報告。
(なお、ダウンロードに関しては、5章を良く参照して下さい。)

- ① レポートの内容レベルは、A4x1枚程度。
- ② 次回の授業開始時に、提出して下さい。
(ただし、それ以前に提出する場合は、メールで願います。
アドレス: fwhy6454@mb.infoweb.ne.jp)

7. 参考書ほか

- 大木幹雄「データベース設計の基礎」(日本理工出版会)
- 林 衛「ERモデルによるデータベース設計技法」(SRC社)
- 小野哲ほか「まるごと図解、SQLがわかる」(技術評論社)
- A. M. デービス「ソフトウェア開発201の鉄則」(日経BP社)
- 村上毅ほか「MySQL活用ガイド」(秀和システム)
- ライアンほか「SQLプログラミング入門」(ソフトバンク)
- <http://www.ann.hi-ho.ne.jp/hirok/sql/index.html>
- <http://www.rfs.jp/sitebuilder/sql/>