

データベースの論理設計技法

Logical Design Method for Database

従来、データベースの設計は、商用のデータベース管理システムが提供する階層形、ネットワーク形、あるいはリレーショナル形の3種類の記述法を用いた論理設計と、ディスク上の格納構造を決定する物理設計の2段階で進められてきた。しかし、従来論理設計には、データ間の関係の問題と性能の問題とを混在して設計してきたところに問題があった。そこで、この論理設計を二つに分け、データ間の関係はエンティティリレーションシップ記法を用いた概念設計で、性能は上記3種類の記述法による論理設計で行なうこととし、全体で3段階の設計法を開発した。また、工数と質の向上のため、ツールのイメージを与えた。本設計法は、実システムへの試用実験により、その有効性を確認し、実用化への見通しを得ている。

近藤秀文* Hidefumi Kondô

酒井博敬** Hirotaka Sakai

小林正和* Masakazu Kobayashi

1 緒 言

データベースを応用するシステムの建設では、その開発及び保守を能率良く行なうため、通常、商用のDBMS(データベース管理システム)を使用する。このデータベース応用システムが、利用者の要求に応じて柔軟かつ迅速にデータを提供するための決め手になるのは、データベース中で管理するデータの構造を設計する作業の良否である。

従来この設計には、次に述べるような問題があった。(1)商用のDBMSでは、データの蓄積の枠組み(論理構造)が、階層形、ネットワーク形、リレーショナル形のいずれか一つの記述法に限定されており、実在するデータを与えられた枠組みに当てはめる仕事には、データベースの専門家を必要としたこと。(2)複数ユーザーの多目的要求に効率良く応じられるように、データベースを設計する仕事は設計者の能力に大きく依存すること。したがって、(3)効率の良いデータベースを設計するためには、利用者のデータ要求をモデル化する段階からデータベースの専門家が必要であること。更に、(4)利用者に理解できるように、データが記述されないままデータベースの専門家によって効率と本来のデータそのものとの問題を共に混在させた設計案が作成されることが多く、利用者の理解が困難で仕様段階でのレビューが十分できず運用時にトラブルを発生しやすいこと。

以上の問題を解決するために、データに対する要求をデータベースの非専門家である利用者にも素直に理解できるデータの記述法に対する要望が高まってきた。この目的で使用する記述法を、ここでは「概念モデル記述法」と呼ぶ。また、ディスクなどの記憶装置上での格納構造を表現するものを、ここでは「物理構造記述法」と呼ぶ。本稿で述べる設計技法は、(1)利用者に理解でき、しかも専門家によるデータベース設計の出発点となる概念モデルの記述法を与え、(2)論理構造及び物理構造への変換あるいは設計は、定型化された技術及びツールを利用して能率良く設計する、という方法である。

2 従来の設計法と新しい設計法

2.1 従来の設計法の問題点と概念モデルの必要性

論理構造の記述法を用いた業務ニーズのデータモデリングには、次のような問題があった。例えば、「得意先と営業所と

の契約記録」を階層形で表わす方法には、図1(a)、(b)、(c)、(d)などがあり、それぞれ特徴がある。同図(a)の構造では、得意先をキーにした営業所及び契約記録の検索は高速に行なえるが、営業所をキーにした検索では全データベースの検索になってしまう。したがって、営業所のところをセカンダリインデキシング法(4章で説明)で直接検索できるようにするか、同図(b)に示すように、営業所を独立させて、ポインタ(4章で説明)で関係を付け、どちらからでも検索できるようにするなどの方法が考えられる。また、営業所データの重複が多くなるので、営業所を独立させ、得意先との関係には営業所コードだけを使用し、更に、得意先と営業所コードを一つにまとめてしまうという方法もある。同図(c)は、この構造を表わしたものである。また、契約記録は最新3回分だけと限定すれば、同図(d)に示すように、得意先と契約記録を一つにまとめることもできる。

以上のような議論は、検索性能及び格納エリア効率に関するもので、設計者が利用者からデータに関する要求を整理する段階で議論するような内容ではない。階層形記法では、

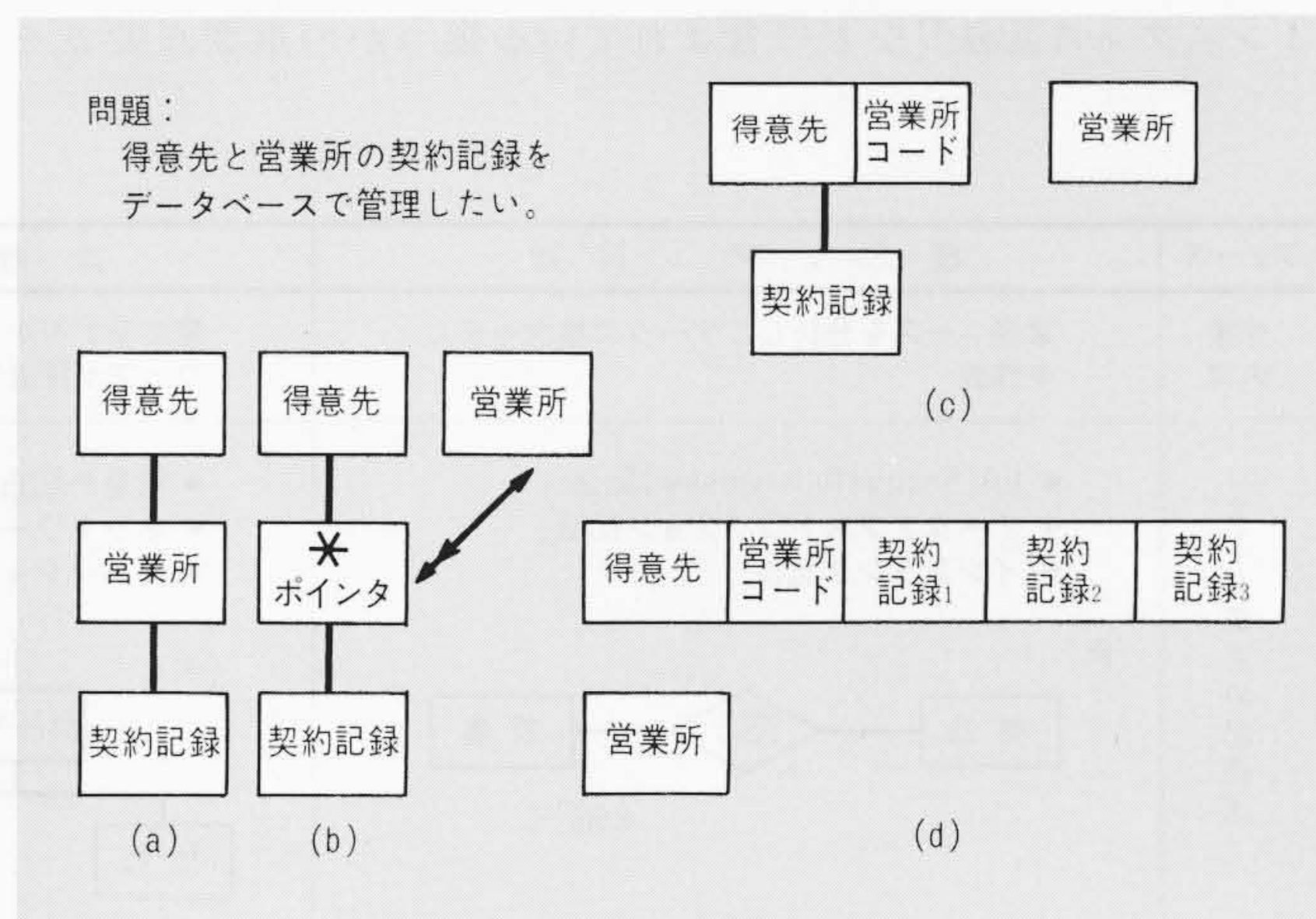


図1 階層形記法を用いた論理構造の種々の表現法 データの内容は同じでも、いろいろな表現法がある。

* 日立製作所システム開発研究所 ** 日立製作所日立技術研修所

データの内容だけに注目し、性能及びエリア効率は切り離して議論するというようなことが本質的に無理である。同様なことは、程度の差はあるにしてもネットワーク形及びリレーショナル形にも言えることである。すなわち、論理構造記述法では、データの内容、性能及びエリア効率を一度に考えねばならず、利用者の要求を議論する段階のデータ記述法としては適切でない。この問題を解決するものが、概念モデル記述法であり、以下に述べるような特長がある。

- (1) 利用者と設計者の共同作業によるデータのモデル化が容易になる。
- (2) 概念モデルは、DBMSに依存しないので、DBMSの選択とは切り離して設計が行なえる。
- (3) 概念モデルから論理構造への基本的な変換は、機械的に行なえる。

図1の論理構造は、概念モデルでは、後述する図3(a)の1種類だけで表わすことができる。

2.2 新しい設計法

図2は、従来及び本稿での設計手順の比較を示したものである。従来に比べて新しい設計では、概念モデルの設計段階が増えるが、業務ニーズのデータモデリングが容易になり、また概念モデル設計後は、ツールを使用して能率良く設計を行なえるので段階的な設計が行ないやすい。同図(a)に示すように概念モデルの設計は、まず利用者あるいは複数の利用者ごとに概念モデル記述法を用いてローカルビュー(後述)を作成し、次にこれらを統合するという手順で行なう。論理構造設計後の物理構造の設計手順は、従来の方法、本稿の方法どちらも同じである。すなわち、本稿の設計法は、概念、論理及び物理の3段階に分かれる。表1は、三つの各段階で使用される代表的なモデル記述法、作業内容及び支援ツールの内容を示したものである。

本稿では、概念モデル記述法としてER(エンティティリレーションシップ)記法¹⁾を改良した記述法を、論理構造記述法として日立製作所DBMSの階層形記法²⁾を、物理構造記述法として階層アクセス法²⁾を用いて説明する。

3 概念モデルの設計

3.1 概念モデル記述法(ER記法)

本節で述べるER記法は、データアブストラクション記法³⁾、インスタンス記法⁴⁾などに含まれている幾つかの重要な要素を

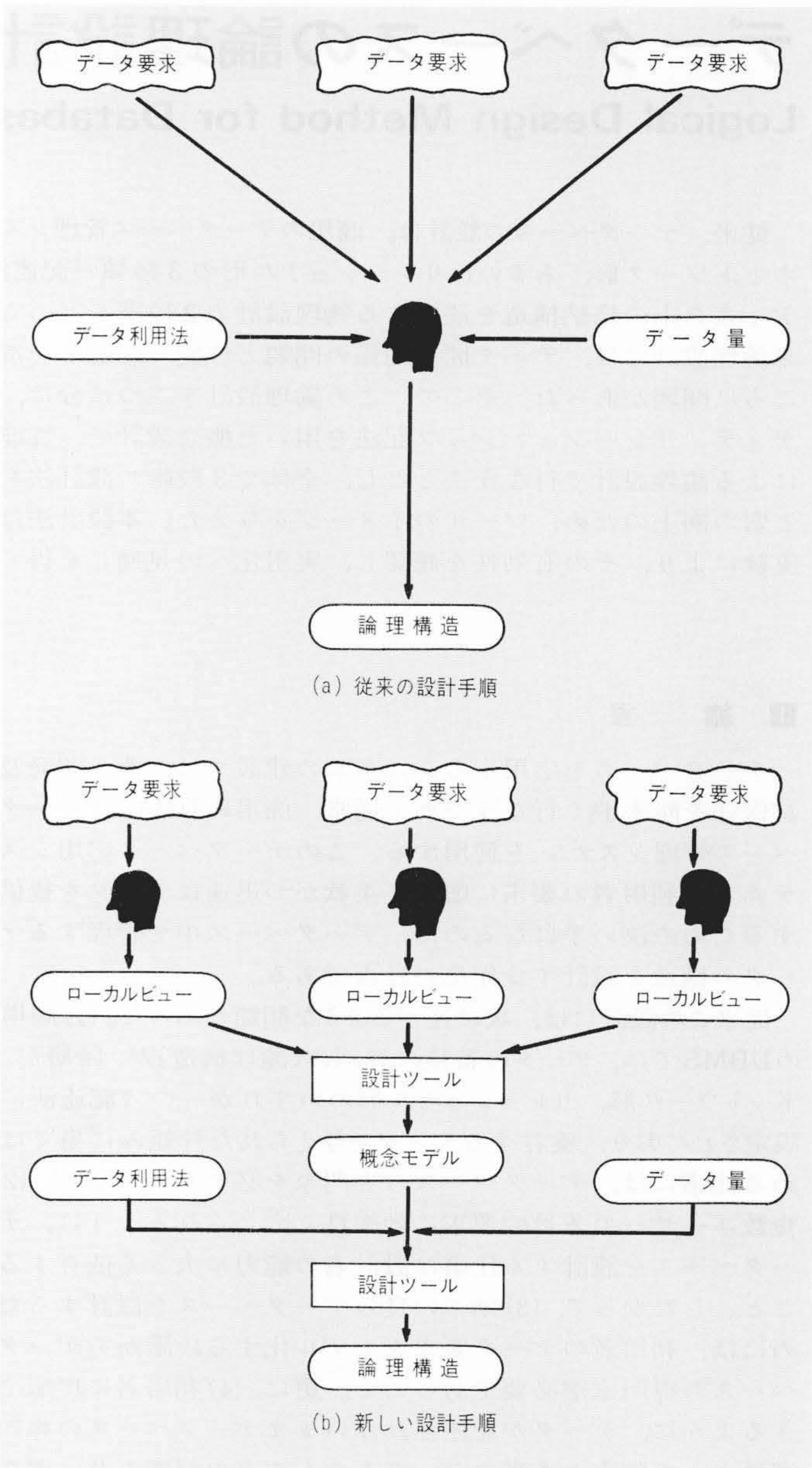


図2 従来と新しいデータベースの設計手順 従来は、商用のデータベースソフトウェアが提供するデータ記述法を直接用いて設計を開始していたため、利用者と設計者の対話がスムーズに行なえなかった。

フェーズ	概念モデル設計	論理構造設計	物理構造設計
作業内容	業務ニーズを分析してデータの概念モデルを作成	概念モデルから商用のデータベースソフトウェアが提供する論理構造へ変換	論理構造のディスク上での格納構造を決定
データ構造の記述法	● ER(Entity-Relationship)記法 ● データアブストラクション記法 ● インスタンス記法 (例) ER記法	● 階層形記法 ● ネットワーク形記法 ● リレーショナル形記法 (例) 階層形記法	● インバーテッド法 ● マルチリスト法 ● 階層アクセス法 (例) 階層アクセス法
支援ツール	業務ごとに会話形式でローカルビューを作成し、業務間データのシノニム(異名同義)、ホモニム(同名異義)の指定によりローカルビューを統合して概念モデルを作成する。	データ利用法、データ量を入力として、適正な論理モデルを出力する。	データ利用法、データ量、ディスクの形を入力として、ディスク上での適正な格納構造及び性能見積値を出力する。

表1 設計フェーズとその内容 設計の各フェーズでは、設計作業に合うデータの記述法、支援ツールが必要である。

加えて改良したものである^{5),6)}。

以下にER記法の主要な4要素について述べる。

(1) エンティティ

はっきり区別のできる実体を言い、個々の人、組織、事象などがエンティティである。同種のエンティティを集めたものをEセット(エンティティセット)と言う。

図3(a)では、得意先及び営業所がEセットを表わす。

(2) リレーションシップ

エンティティ間の関係を言い、ある特定の得意先 e_i (エンティティ)がある特定の営業所 f_i (エンティティ)と契約しているとき、この両エンティティの関係、すなわち両者の組(e_i, f_i)をリレーションシップと言う。同種のリレーションシップを集めたものを、Rセット(リレーションシップセット)と言う。同図(a)の契約はRセットを示す。同図中の n, m はマッピングレシオと言い、一つの得意先が n 個の営業所とリレーションシップをもち、逆に一つの営業所が m 個の得意先に対応していることを示す。これは重要な数値であり、もし n でなく1ならば、どの得意先も一つの営業所とだけ契約を結ぶということを表わしてしまう。

(3) アトリビュート

エンティティ及びリレーションシップに関する属性を表わす。例えば、得意先には、名称、住所などがある。契約には契約年月、契約内容などがあるが、これらは、得意先や営業所に固有なものでなく、得意先と営業所の組に対して意味をもつという点に特徴がある。

(4) カテゴリー

Eセット中のエンティティ、あるいはRセット中のリレーションシップをある基準で分類したり、同種のEセットあるいは同種のRセットを統合する機能である。図3(b)の取引先分類はカテゴリーを表わす。このカテゴリーは、取引先を納入先と仕入先に分類する役割を果たす。カテゴリーは、二つの利点をもつ。第1は、取引先全体に対するアトリビュートは取引先に、納入先あるいは仕入先だけに特有なアトリビュートはそれぞれ納入先あるいは仕入先に定義すればよい。第2

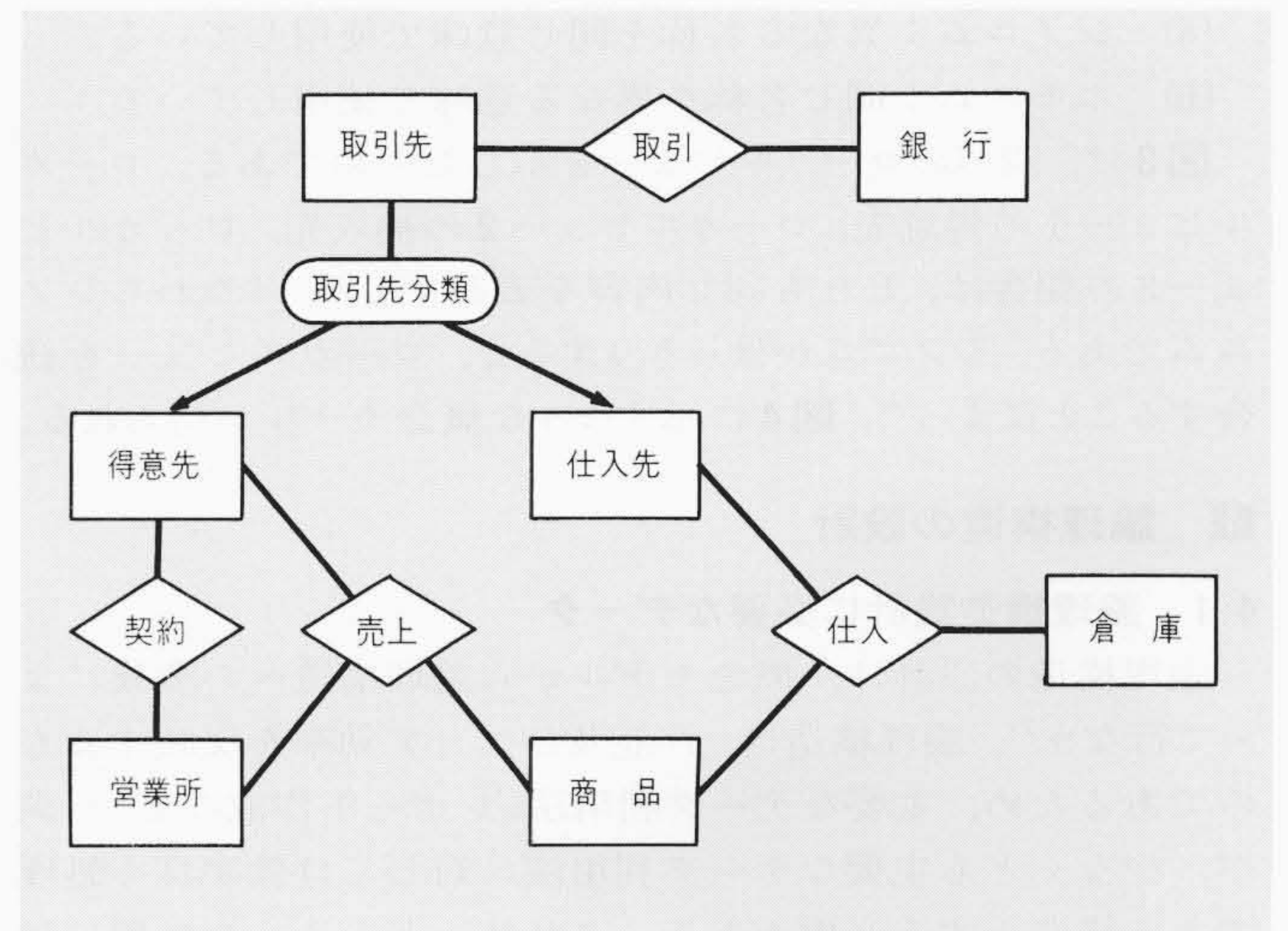


図4 概念モデル ローカルビューを統合して、概念モデルを作成する。

は、他のEセットとの関係を、取引先全体に対するものとして、あるいは納入先、仕入先だけとのリレーションシップとして定義できることである。図3(b)で、銀行との取引は、納入先及び仕入先のいずれにも関係するものであるから、取引先と銀行との関係として定義してある。

3.2 概念モデルの設計手順

利用者のデータ要求を概念モデルにまとめる作業は、規模が小さい場合には、一度に概念モデルを書くことが可能であるが、大規模の場合には、次の二つのステップで作成するほうが作りやすい。

- (1) 利用者あるいは利用者グループ別にデータ要求をER記法で記述する。これを、ローカルビューと言う。
- (2) ローカルビューを統合して概念モデルを作成する。

ローカルビューを統合する段階で発生する問題は、各ローカルビューで名称の不統一が起こることである。名称の不統一には次の種類がある。

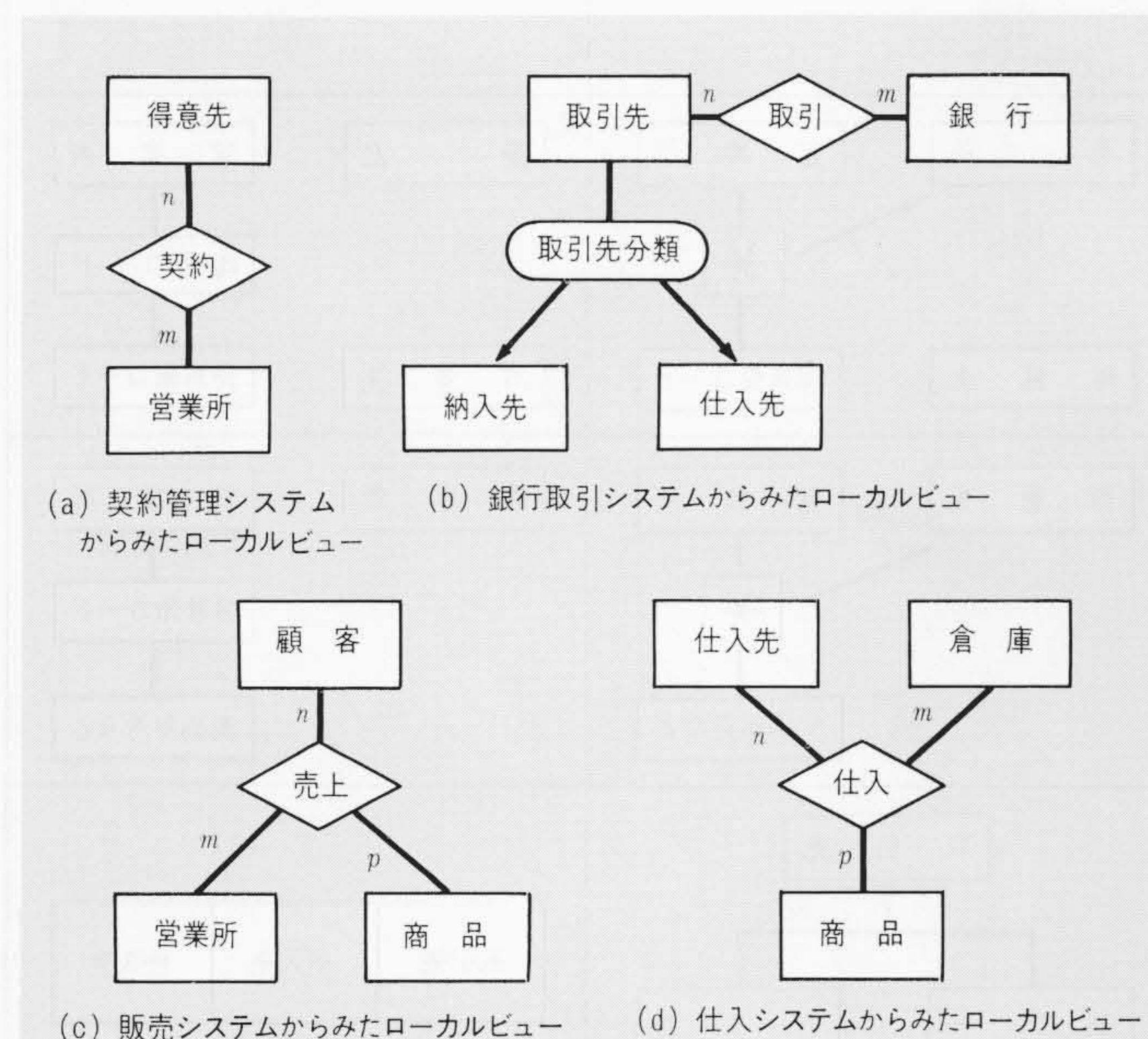


図3 ローカルビュー 利用者あるいは複数の利用者ごとに、データベースに対する要求をER記法で記述する。これをローカルビューと言う。

E セット	得意先	500件
E セット	営業所	10件
E セット	商品	300件
R セット	売上	5,000件
関係Eセット	得意先(平均	10件)
	営業所(平均	500件)
	商品(平均	17件)

(a) データ量

- (1) ある営業所が契約している得意先リスト
40回/月 検索 営業所(営業所コード) KEY
→契約(得意先コード) ALL
- (2) ある商品の営業所別売上リスト
20回/月 検索 商品(商品コード) KEY
→売上(営業所コード, 数量, 金額) ALL

(b) データ利用法

図5 データ量とデータ利用法 エンティティ及びリレーションシップに関するデータ件数、データ処理内容・頻度を指定する。

- (a) シノニム：異なる名称を同じ意味で使用している。
- (b) ホモニム：同じ名称を異なる意味で使用している。

図3は、四つのローカルビューを示したものである。ローカルビュー1の得意先、ローカルビュー2の納入先、ローカルビュー3の顧客は、どれも同じ内容を表わすもの、すなわちシノニムである。シノニムがはっきりすると、ローカルビューを統合することによって、図4に示すような概念モデルが得られる。

4 論理構造の設計

4.1 論理構造設計に必要なデータ

論理構造の設計は、概念モデルから論理構造への変換によって行なう⁷⁾。論理構造は、性能及びエリア効率を反映するものであるため、主要なデータ利用法(レポート作成など)を調べ、少なくとも主要なデータ利用法に対しては効率良く処理できる構造とする必要がある。これは、どのような処理に対しても効率良く処理できるようにしておくなどということは、實際上不可能なためである。図2(b)で示したように、論理構造設計では、データ利用法及びデータ量を使用する。このデータは概念モデルを対象にして記述する。図5(a)は、データ量の記述例である。同図で、「関係Eセット得意先(平均10件)」は、マッピングレーショの具体的な数の記述であり、ある特定の得意先の売上データが平均10件であることを表わす。同

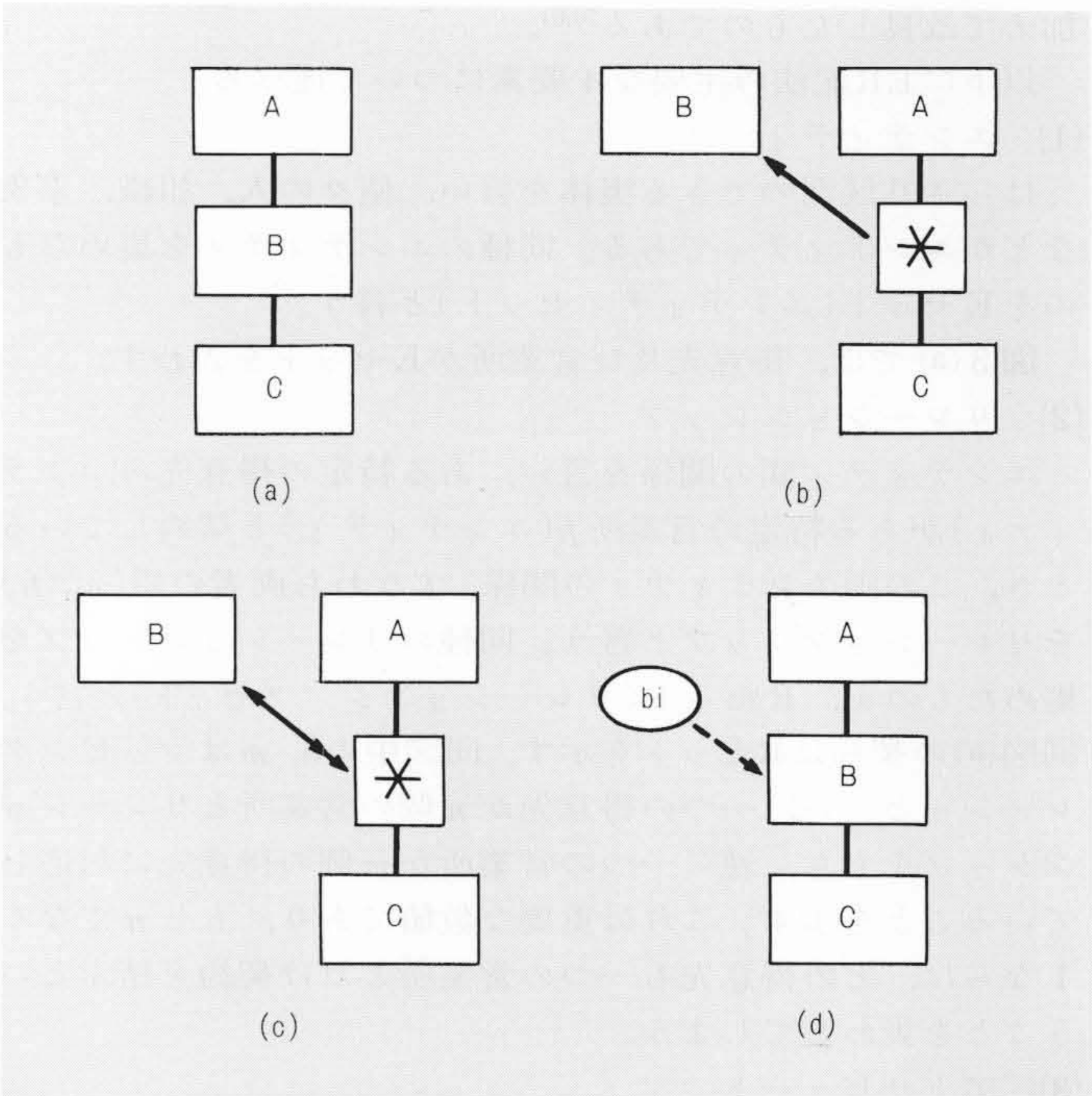


図6 階層形記法の基本形 階層形では、同じ内容のデータを表現する方法はいろいろあるが、性能・エリア効率が変わる。

区分	E R 記 法		階 層 形 記 法	
(a)	マッピングレーショ $n:1$			
(b)	マッピングレーショ $n:m$			
(c)	三つの エンティティ間の リレーションシップ			
(d)	リレーションシップ のATTRIBUTE			
(e)	カ テ ゴ リ ー			

図7 基本変換規則 概念モデルから論理構造への変換では、ER記法から階層形への変換規則を使用する。

図(b)は、データ利用法を示すデータである。同図中の(1)は、得意先リストの出力が毎月40回あり、その処理は営業所コードを検索キーとして、それが契約している得意先コードを検索する処理であることを示す。

概念モデルから論理構造への変換は、概念モデル、データ利用法及びデータ量に基づいて実行する。以下に、まず階層形記法の概略を述べ、次に変換法について説明する。

4.2 論理構造記述法(階層形記法)

階層形記法では、一つ以上のフィールド(データアイテムのこと)から成るものをセグメントと言ひ、これを用いて木構造の形を作る。図6は、階層形による基本形を示したものである。階層形では、最上段のセグメントをルートセグメントと言ひ、ルートセグメントは、キー検索が可能である。同図中のA、同図(b)、(c)中のBがルートセグメントである。＊はポインタから成るセグメントであり、次に述べるような機能をもつ。同図(b)の場合には、Aの下のポイントをとってBを見ることが可能である。したがって、この構造はBデータの重複がないという点を除いて、同図(a)と全く同じ内容をもつものである。同図(c)の場合には、(b)と同様な機能以外に、Bのほうから逆にポイントをたどってAを見ることが可能である。同図(d)は、セカンダリインデキシングと呼ぶもので、セグメントBの中の特定のフィールドbiをキーにした検索を可能にする特殊な機能である。論理構造は、これらの基本形を組み合わせで作成される。

4.3 概念モデルから論理構造への変換

概念モデルから論理構造への変換では、まず、ER記法のマッピングレーショを用いた変換を行ない、次にデータ利用法及びデータ量を参考にした見直しを行なう。図7は、主としてマッピングレーショを用いたER記法から階層形への基本変換規則を示したものである。同図(a)は、マッピングレーショが $n:1$ の場合の変換であり、同図(a)で示した基本形に変換される。同図(b)は、 $n:m$ の場合の変換であり、最も基本的な変換として同図に示すような二つの形がある。同図(c)は、三つのエンティティ間の関係をもつ構造の変換、同図(d)

は、リレーションシップの属性の変換、同図(e)は、カテゴリーの変換である。これらの変換された構造は、いずれも図6で示した階層形の基本形に対応する。

図8は、データ利用法及びデータ量に基づく基本的な変換規則であり、利用法や量の観点から構造の見直しを行なうことを目的とする。同図(a)は、営業所をキーとした検索以外に得意先をキーとした検索をも可能にするため、図6(d)のセカンダリインデキシングの基本形を利用した構造である。図8(b)も、同じく営業所及び得意先の双方向からの検索を可能にするための構造であるが、図6(c)の基本形を利用した構造である点異なる。図8(c)は、データ量に基づく変換規則である。リレーションシップ契約のマッピングレーショの具体値をみると、特定な営業所に対する契約得意先数は500件であるのに対して、逆は3件である。ルートセグメントの下に結合されるセグメントの数を少なくしたい場合には、図8(b)よりも(c)のほうが好ましい。

ER記法から階層形への変換は一意的でないため、複数の論理構造の候補がでてくる。適切な候補の選択は、性能、エリア効率、見やすさ、保守のしやすさ、などの観点から決定する。性能及びエリア効率は、5章で述べる物理構造設計後のデータを用いると、より高い精度で見積値を算出できるので評価しやすい。図9(a)は、図4の概念モデルに対する論理構造の一つの候補である。図9(a)の論理構造の各セグメントに対して、フィールドのレイアウトを決めると論理構造設計は終了する。図9(b)にセグメントレイアウトの例を示す。

5 物理構造の設計

ディスク上での格納構造の設計については、従来から多くの議論がなされ、重要な項目や設計方法も相当明確になっているので^{8),9)}、技術的な要点だけについて述べる。

ADM(Adaptable Data Manager)²⁾の物理構造設計上の重要な設計項目には、下記のものがある。

(1) インデックス構造を用いて、キーの大きさの順によるデータ検索を可能にするか否か。

区分	E R 記 法	階 層 形 記 法
(a)	得意先をキーとする検索あり。 マッピングレーショ $n:1$	
(b)	得意先をキーとする検索あり。 マッピングレーショ $n:m$	
(c)	リレーションシップ 契約 関係のエンティティ 営業所 (平均500件) 得意先 (平均3件) 営業所から得意先への件数は多い。	

図8 データ利用法及びデータ量に基づく変換規則 純粋なデータの構造だけでなく、データ利用法やデータ量を参考にして論理構造を決める。

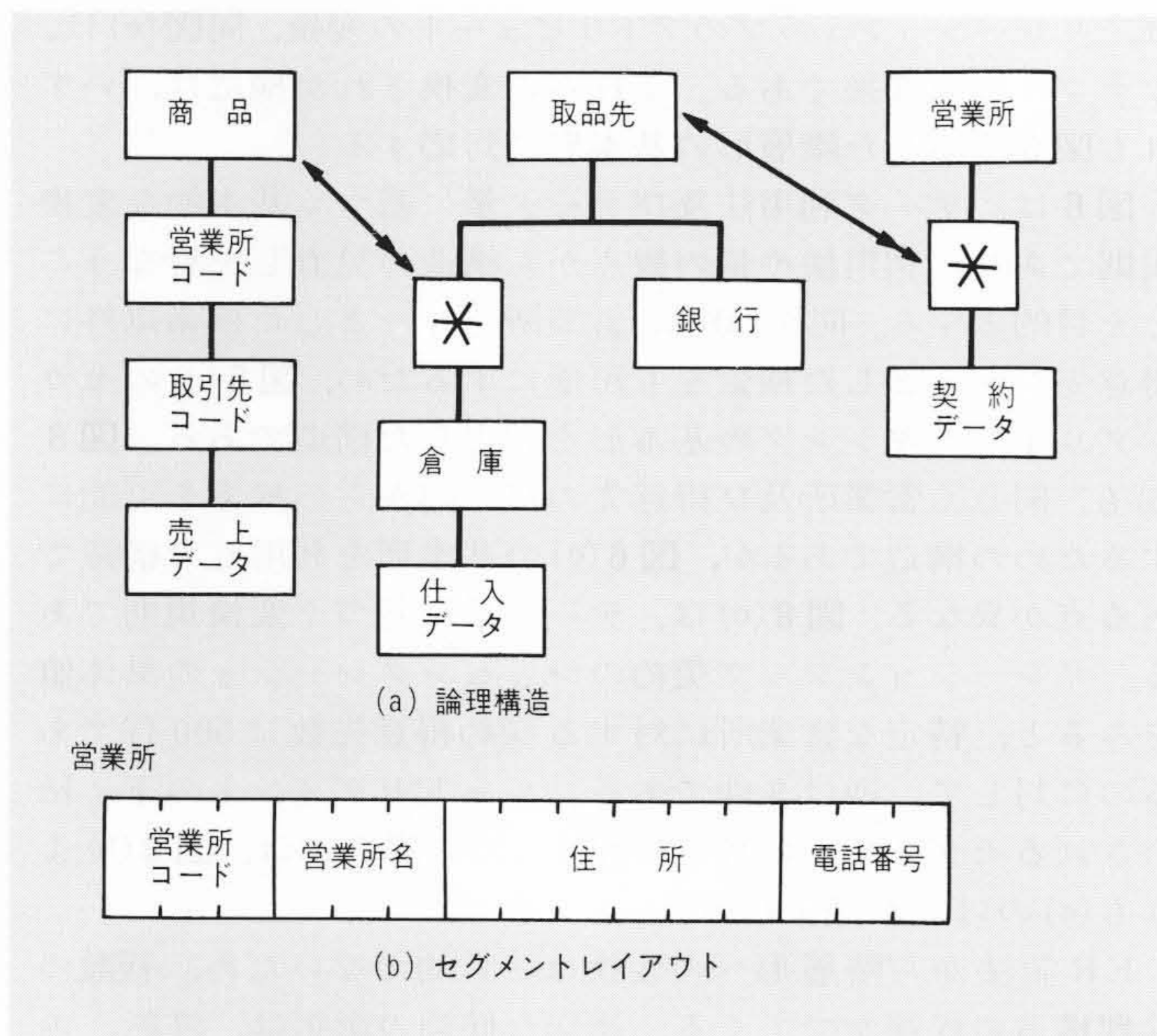


図9 論理構造とセグメントレイアウト 階層形を用いた論理構造及びセグメント内のフィールドのレイアウトである。

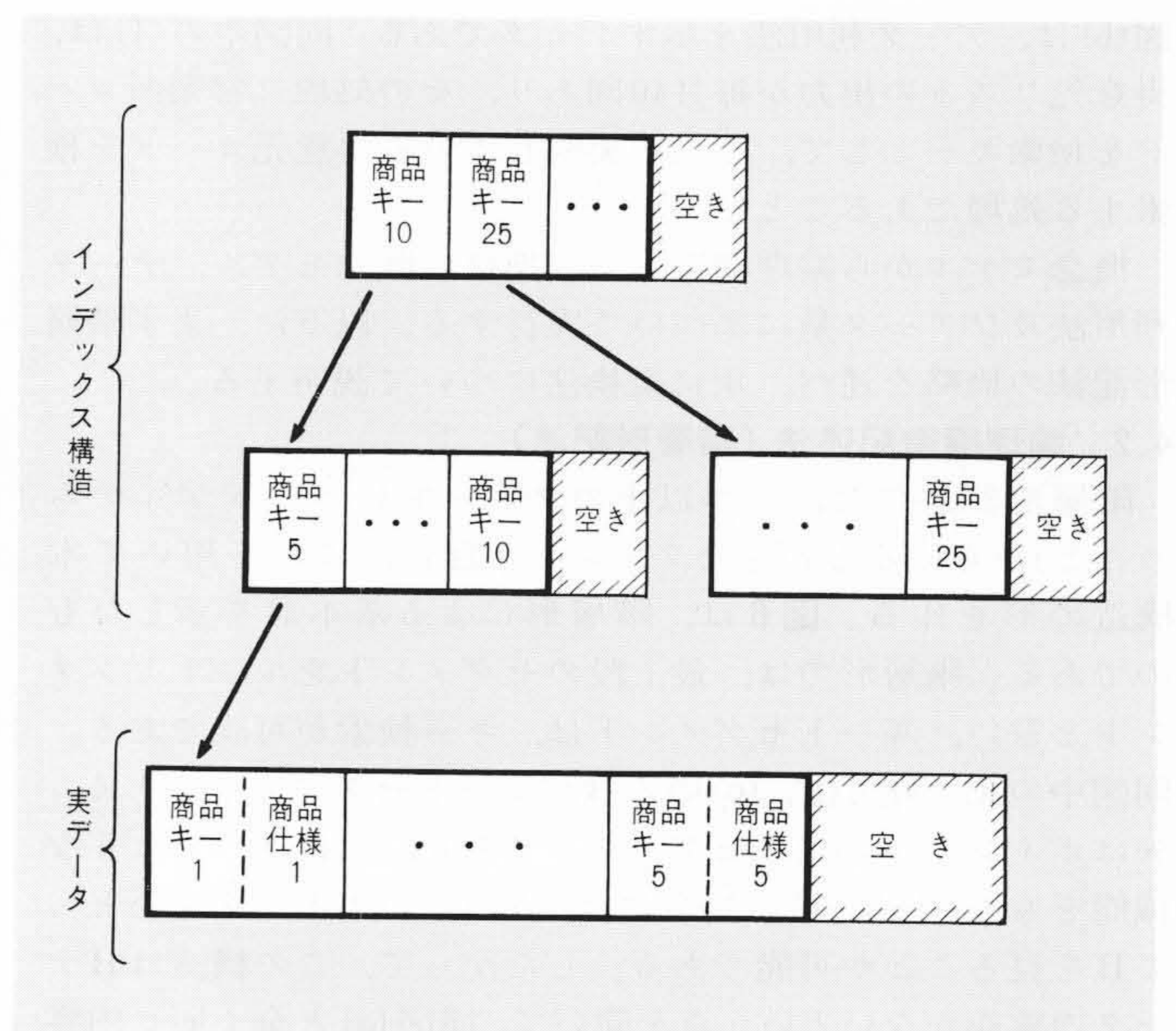


図10 インデックス構造を用いた物理構造 インデックス構造を用いると、キー値の大きさの順にデータの検索が行なえる。

- (2) ハッシュ構造(キー値を関数で格納アドレスに変換する方法)により、高速なキー値検索を可能にするか否か。
- (3) 追加データ用空きエリアを、どの程度、どの場所に、事前に確保しておくか。
- (4) データの格納ブロックサイズ、データ間の細かいポインタをどのように設定するか。

図10は、インデックス構造を採用したときの物理構造の例である。

6 設計支援ツール

第3、第4章及び第5章に述べた設計法に従ったツールとして実験的にDBDS(Data Base Design Support system)があり、次の機能をもつ。

- (1) DBDS/CSDA(DBDS/Conceptual Schema Design Aid)
図形端末を用いて対話形式でローカルビュー及び概念モデルを作成する^{6),10)}。図3を入力すると図4を出力するツールである。
- (2) DBDS/LSDA(DBDS/Logical Schema Design Aid)
概念モデルから論理構造への変換ツールであり、自動設計及び設計者による変更機能をもつ。概念モデル(図4)、データ量及びデータ利用法(図5)を入力すると論理構造[図9(a)]を出力するツールである。
- (3) DBDS/AIDE(DBDS/ADM Information Design Package)
論理構造から物理構造への変換ツールであり、図9及び同図に対するデータ量、データ利用法を入力すると格納構造を決定し、ディスク容量及び性能の見積値を出力する⁸⁾。

7 結 言

データベースを応用するアプリケーションシステムを対象としたデータベースの設計技法及び実験用ツールについて述べた。本稿の設計法は、まず(1)概念モデルを作成し、次に(2)商用DBMSの提供する論理構造に変換し、最後に(3)ディスク上の物理構造を決定する、という3段階設計法であり、次に述べるような特長がある。

- (1) 利用者と設計者が共同で行なうデータモデリング作業が容易である。

- (2) 概念モデル記述法で表現された概念モデルに基づき、論理構造及び物理構造の代案作成を自動的に行なえるため、設計工数が削減される。

ある実システムへの本技法・ツールの適用事例では、約30%の工数節減が得られた。今後は、システム、ソフトウェア、プログラムの各設計との関連を明確にし、一貫した設計システムの中で機能するツールへの展開を図る計画である。

参考文献

- 1) P. Chen: The Entity-Relationship Model: Toward a Unified View of Data, ACM Trans. on Data Base Systems, 1, 1, 9~36 (1976)
- 2) データマネジメントシステム ADM システム管理者ガイド システム設計編(マニュアル番号: 8080-6-202-60)
- 3) J. M. Smith, et al.: Database Abstraction: Aggregation and Generalization, ACM Trans. on Data Base Systems, 2, 2, 105~133 (1977)
- 4) S. Navathe, et al.: View Representation in Logical Database Design, ACM-SIGMOD International Conf. on Management of Data, 144~156 (1978)
- 5) H. Sakai: A Method for Defining Information Structures and Transactions in Conceptual Schema Design, 7th International Conf. on Very Large Data Bases 225~234 (1981)
- 6) 近藤, 外: データベース設計における概念スキーマ生成支援ツール, 情報処理学会第23回全国大会, 495~496 (1981)
- 7) H. Sakai: A Unified Approach to the Logical Design of a Hierarchical Data Model, International Conf. on Entity-Relationship Approach to System Analysis and Design, 73~88 (1979)
- 8) H. Kondo, et al.: A Database Design Tool for Hierarchical Databases, IEEE COMPSAC 80, 533~538 (1980)
- 9) 近藤, 外: 階層構造データベースの性能に関する解析的見積り法, 情報処理学会論文誌, 21, 5, 418~425 (1980)
- 10) H. Sakai, et al.: A Development of a Conceptual Schema Design Aid in the Entity-Relationship Model, 2nd International Conf. on Entity-Relationship Approach (1981)