データベース・システム 45－1
（１９８５．１．２２）

データベース設計システム DBDESIGN

中野 勝之　池田 幸雄　山多 昭　馬場 正和
（日本電信電話公社 横須賀電気通信研究所）

１．はじめに

データベース設計では，データベース化する業務で扱う情報の理解とその情報の形式的表現，使用するデータベース管理システムの仕様に基づいたデータ構造の記述とその最適化等の多種類の設計作業がある。このため，データベース設計には様々な知識が必要であり，データベースの大規模化，複雑化に伴って，データベース設計は増々困難なものとなっている。その対策として，体系的かつ具体的なデータベース設計の方法論の確立や設計支援ツールによる機械支援が強く要求されている。

データベース設計の概念的枠組は，①要求分析，②概念設計，③論理設計，④物理設計から構成され[1]，各工程の設計の方法論，各種支援機能が提案されているが，以下の問題が残されている。

（１）概念設計作業は，ビュー設計とビュー統合に大きく分割される。データベース化対象世界を構成する対象は，多くはビュー設計時に認識されるが，認識の仕方，詳細さのレベルがビュー毎に異なることから，ビュー統合時にも新しい対象が認識される。C.Batini等[2]は，実体関連モデル[3]を拡張したデータモデル上で，２つの対象に付けられた名前の競合やそれらの両立性の問題を詳細に分析している。また，ビュー統合方法として①ビュー競合分析，②マージ，③改良の３ステップに分割した方法を提案し，ビュー競合分析では，２個のビューの比較による上述の分析と問題点の解決，マージでは対象の機械的な合併を行う。改良では，ビュー間の間隙の解析による新しい対象の認識，冗長性の解析，データ構造の明確さ，簡明さを向上させるための再構成を行う。また，S.B.Navathe等[4,5]は，実体関連モデルを拡張したE-C-Rモデル上で関連と実体の統合方法を分析している。このビュー統合法は，①ビュー間の実体，関連等の対応をとるビュー統合の前処理，②統合，③ビューとグローバル・ビューのマッピングの生成の３ステップに分解している。クラス間の包含関係，ロール等の場合分けを行い，関連と実体の統合方法を詳細に分析している。しかし，いずれも対象の重なりの分析から新しい対象が認識されることが考慮されていない。

（２）論理設計におけるCODASYL型[6]の論理スキーマ生成法には，P.P.Chenの実体関連モデルから論理スキーマを生成する方法[3]やK.B.Irani等の二項モデル等から有限分岐法を用い性能的に最適な論理スキーマを生成する方法[7]がある。しかし，CODASYL型論理スキーマの制約条件の生成法が示されていない。また，後者はデータ項目数が多いと膨大な計算機時間を必要とし，大規模データベースの設計に対しては実用的と言えない。

（３）物理設計における記憶スキーマの生成法については，H.K.Jainが，記憶スキーマの生成問題を目標計画法問題に還元することにより，最適記憶スキーマ生成法を提案している[8]。しかし，詳細な目標値の設定が必要であり，この設定を適切に行うことが困難である等の問題がある。DBDE[9]は，データベースの性能値を設計者に提供することにより，性能面からスキーマを修正することを狙いとしている。しかし，出力される性能値がI/O時間，CPU時間，レスポンス・タイム等のマクロな情報であるため，修正箇所を特定できないという問題がある。

本稿では，筆者らが開発したCODASYL型データベースの設計支援を目的とするデータベース設計システム（DBDESIGN）について，概念設計の方法論，論理設計における論理スキーマの生成法，物理設計における記憶スキーマの生成法と性能面からのスキーマ修正法について述べる。提案する主な内容は以下の通りである。第一点として，２つの対象の重なり方に着目して，２つの対象の関係を整理する。また，C.Batini等のビュー競合分析を２ステップに分け，ビュー統合を４ステップに分解した概念設計の方法を提案する。第二点として，概念スキーマから性能の良い論理

スキーマを生成する方法を示す。また，制約条件の生成方法についても述べる。第三点として，大規模データベースへの適用を目的としたアクセス効率の良い記憶スキーマの生成方法を示す。第四点として，スキーマの性能改善に利用できる分析情報として，業務プログラムのデータベースへのアクセス特性とデータベース中のデータの分布特性の詳細を収集する方法を示す。

本稿の第二章では，データベース設計工程の概念的枠組みとDBDESIGNの支援範囲を述べる。第三章では，方法論を中心に概念設計について述べる。第四章では，CODASYL型論理スキーマの生成法を中心に論理設計について述べる。第五章では，記憶スキーマの生成法と性能面からスキーマの修正を可能とする性能分析データの収集法を中心に物理設計について述べる。最後の章で今後の課題について述べる。

2．データベース設計工程とDBDESIGNの支援範囲

2．1　データベース設計工程の枠組み

データベース設計は以下の４工程に分割される。

（１）要求分析

利用者のデータベースへの要求を収集，分析し，データベースの設計条件を決定する工程である。データ要求，処理要求等を分析，整理する。

（２）概念設計

概念モデルを用いてデータベース化する業務のデータ構造（概念スキーマ）を設計する工程である。個々の業務毎のビューを作成するビュー設計と，それらを統合し概念スキーマを作成するビュー統合の２段階で設計する。

（３）論理設計

概念スキーマを基に，データ量，アプリケーション・プログラム（ＡＰ）のアクセス特性等の性能やデータベース・システムの運用形態を考慮し，インプリメントに用いるデータベース管理システムのデータモデルに従ったデータ構造（論理スキーマ）を設計する。

（４）物理設計

論理スキーマで定義されたデータ構造のデータベース中での物理的な構造（記憶スキーマ）を設計する。また，データベースの性能を評価し，論理／記憶スキーマを改良する。

2．2　DBDESIGNの支援範囲

DBDESIGNは次の支援機能を提供する。

（１）概念設計工程……概念スキーマの記述言語（ＩＳＬ），ビュー・マージ機能

（２）論理設計工程…… CODASYL型論理スキーマの生成機能

（３）物理設計工程……記憶スキーマの生成機能，性能分析データ収集機能

3．概念設計

3．1　概念スキーマ記述に用いるデータモデル

概念スキーマは対象世界を自然に表現できる実体関連モデルにより表現する。以下に実体関連モデルを説明する。

（１）実体，実体型……データベース化対象世界を構成する対象のうち，他とはっきり区別できる物や概念を実体，その同種の集合を実体型と呼ぶ。実体型は固有の名前を持つ。

（２）関連，関連型……データベース化対象世界を構成する対象のうち，実体間の関係を表す情報を関連，その同種の集合を関連型と呼ぶ。関連型は固有の名前を持つ。

（３）ロール……関連において，関係する実体が果す役割である。

（４）属性……実体／関連の性質を表す情報である。属性は実体／関連内で固有の名前を持つ。

属性が取る値として，文字列／数値の区別と桁数を定義する。

（５）キー……実体，関連を識別することのできる属性である。

（６）数量関係……ある実体型に属する実体がある関連型の幾つの関連に関係するかの表現で

このデータモデルに基づいたＩＳＬの構成を図１に，ＩＳＬによる記述例を図２に示す。

```
SYSTEM IS システム名.
  PROCESS IS プロセス名.
    ENTITY IS 実体名.                          <実体の定義項目>
      KEY IS 属性名 [, 属性名…] .  ……キーの定義
      :
      ATTRIBUTE IS 属性名.          ……属性の定義
      :
    :
    RELATIONSHIP IS 関連名.                    <関連の定義項目>
      KEY IS …                     ……キーの定義
      ATTRIBUTE IS …               ……属性の定義
      OBJECT IS 実体名             ……関連に関係する実体の指定
        [ ROLE IS 役割. ]          ……ロールの指定
        CARDINALITY IS { 1 / C / M } .   ……数量関係の定義 { 1：1 固定数 / C：0 または 1 / M：不定数
      :
    :
  :
```

図１　ＩＳＬの構成

```
PROCESS IS ビュー１.
  ENTITY IS 学校.
    KEY IS 学校名.
    ATTRIBUTE IS 学校名
      TYPE IS CHARACTER 20.
  ENTITY IS 教師.
    KEY IS 名前.
    ATTRIBUTE IS 名前
      TYPE IS CHARACTER 20.
    ATTRIBUTE IS 住所.
      TYPE IS CHARACTER 20.
  ENTITY IS 高校生.
  :
  RELATIONSHIP IS 勤務.
    ATTRIBUTE IS 赴任日.
      TYPE IS CHARACTER 6.
    OBJECT IS 教師
      CARDINALITY IS 1.
    OBJECT IS 学校
      CARDINALITY IS M.
  RELATIONSHIP IS 教育.
  :
```

図２　ＩＳＬによるビュー記述例

３．２　概念スキーマ設計のアプローチ法

概念設計は，ビュー設計とビュー統合に大きく分割される。ビュー統合時には，概念設計における次のような特性を考慮する必要がある。

(1)ビュー設計終了時にデータベース化対象世界の対象が正確に認識されているとは限らない。対象の認識は，設計作業の進展に伴って正確になっていく。

(2)設計者によって，対象の認識の仕方，詳細さのレベルに相違がある。

(3)設計者は，常識等対象に関する様々な知識を持ち，対象やその関係を暗黙のうちに理解する。この結果，どの対象をビューに含めるかの判断は設計者により異なる。

このため，ビュー統合時にも新しい対象が認識されることがある。例えば，各ビューに含まれる対象間に重なりがある場合，この重なりの分析から新しい対象が認識される。

筆者等は２つの記述の関係を，記述が表す対象の重なり方を重視して表１のように分類した。また，各々の解決策の例を合わせて示す。抽象関係，導出関係は２つの記述が表す対象の重なり方が特別である場合として分類される。

表１　２つの記述の関係が問題となるケースとその対策の例

分　類	問題となるケース		対策の例
記述が表す対象が一致する.	名前が異なる（シノニム）		名前を合せる.
	実体／関連における属性の不一致		両方の属性を持つ実体／関連と置換える.
	実体／関連におけるキーの不一致		両方のキーの全てをキーとする.
	表現に用いられている構成要素の不一致		表現を一致させる.
記述が表す対象に重なりがある.	特殊ケース	抽象関係がある. ・集合化 ・汎化 ・集約 導出関係がある ＜タイプ・レベル＞ ・ブール代数 ・関係代数 ＜オカレンス・レベル＞ ・プログラミング言語	可能な限り関係を保存できる関連により表現する.
	一般ケース		新しい対象を見出す等の検討が必要.
記述が表す対象が異なる.	名前が一致する（ホモニム）		正しい名前を付ける.

一方，設計者は非冗長性や正確さの観点から，概念スキーマに含める対象の分析と選択を行っている。新しい対象の認識は，この分析，選択作業の中で行われる。C.Batini等は，ビュー競合分析においてこのような分析と選択を同時に行うとしているが，同種の対象が複数あるとき，2個のビュー間の分析では設計者に誤った選択を強いる場合がある。したがって，競合分析ステップを，対象の分析，問題点の抽出，整理を行うビュー間概念分析ステップと対策案の検討，選択を行うビュー間概念調整ステップに分割することが有効である。すなわち，ビュー統合を①ビュー間概念分析，②ビュー間概念調整，③マージ，④改良の4ステップに分割する。

3．3　設計手順

以下の5ステップの手順で概念スキーマを作成する。

＜ステップ1＞ビュー設計

データベース化対象世界のデータ構造を明確にする第一ステップとして，実体関連モデルにより，各業務のビューを正確かつ詳細に記述する。

＜ステップ2＞ビュー間概念分析

2個のビュー間で，ビュー内の対象の妥当性を分析する。2つの記述が表す対象の関係の有無や対象の整合性，名前付けにおける競合の有無等を分析し，問題点を整理する。

図3は，設計した3個のビュー間に重なりがある場合の例を示す。

＜ステップ3＞ビュー間概念調整

新しい対象の認識等も行い，各問題点毎に不整合の対策案を検討し，局所的に良い解を選択し，ビューを修正する。

図3のビューの修正例を図4に示す。

＜ステップ4＞マージ

ビューを1個のデータ構造にまとめる。

図4の修正したビューのマージ結果を図5に示す。

＜ステップ5＞改良

データベース化対象世界の全体を見通した観点から，冗長表現の削除，単純化等の改良を行い，概念スキーマとして完成させる。

例えば，図5のマージ結果において，関連『通学』は他の関連や実体から導出できるので冗長な関連と言える。

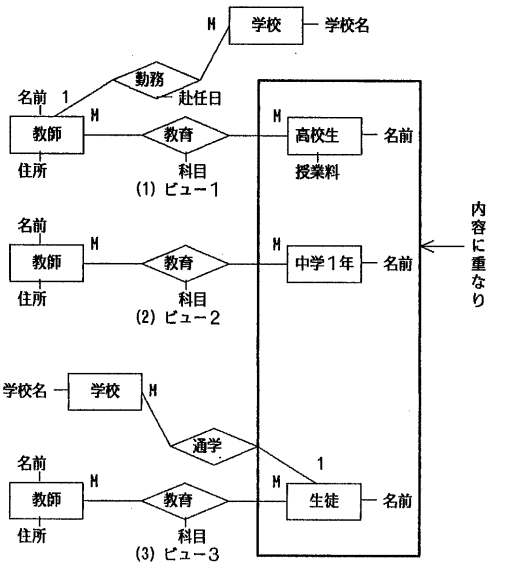


図 3　概念分析の例

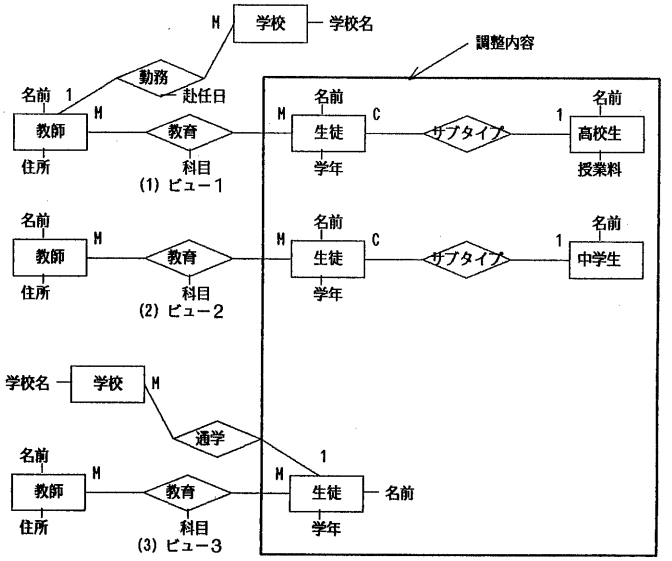


図 4　概念調整の例

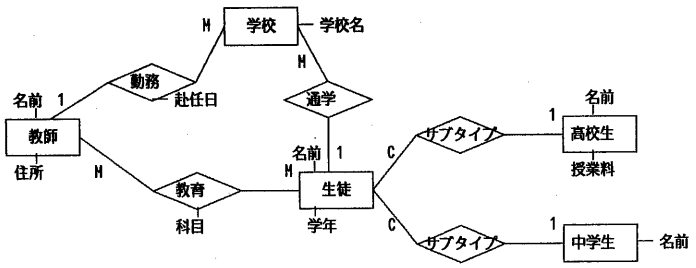


図 5　マージの例

4．論理設計

4．1　設計手順

概念スキーマにおけるデータ構造は，個々のデータ間の複雑な関係を正確に表現することを目的として設計される。これに対し，論理スキーマのデータ構造は，アクセス効率等を考慮する必要があるため，データの冗長配置等により単純なデータ構造となるような工夫がなされている。したがって，性能条件が厳しいシステムを対象とする場合は，概念スキーマにおけるデータ構造をあらかじめ単純化しておくことが必要である。

以上のことから，論理スキーマは以下の手順で設計する。

＜ステップ１＞概念スキーマの表現方法の変更

変更の候補を以下に示す。

①実体間の対応が多対多となる関連型やｎ項関連型による表現は，１対多の２項関連による表現に変更する。これは，データベースの創成／再編成／再構成時間の短縮が狙いである。

②１つの実体型が多くの情報を含むように，実体型の統合を行う。これは，レコードへのアクセスに伴う I/O回数の削減が狙いである。

＜ステップ２＞変更した概念スキーマから論理スキーマの生成

変更した概念スキーマに表現されているデータ構造を基に 4.2節で示す方法により，論理スキーマを生成する。

＜ステップ３＞論理スキーマの完成

概念スキーマから生成できない親子集合順序の追加やその他の必要な修正を行い，論理スキーマを完成する。

4．2　論理スキーマの生成法

CODASYLモデルはレコード，親子集合，制約条件（親子属性，親子集合選択，親子集合順序）の定義等からなる。概念スキーマから論理スキーマを生成する方法を以下に示す。なお，レコード，親子集合の生成方法についてはP.P.Chen[3] により示されているので，それを適用する（下記の①，②）。

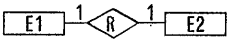
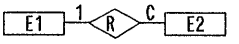
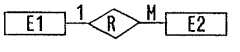
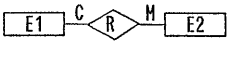
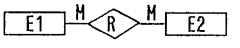
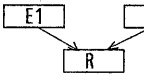
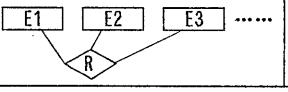
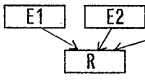
①レコード

実体型に対応してレコードを生成する。属性はこのレコードのデータ項目とする。

②親子集合

属性を持たず，実体間の対応が多対多とならない２項関連型に対応して親子集合を生成する。親／子レコードは数量関係により決定する。その他の関連（実体間の対応が多対多となる２項関連，ｎ項関連型，属性を持つ関連型）に対しては，構造レコードを生成し，これを子レコード，関係する各実体型を親レコードとする親子集合を生成する。関連型の属性はこの構造レコードのデータ項目とする。

表２　親子集合／親子属性の生成方法

関連の分類	ケース	関連の定義（実体関連モデル）	生成するデータ構造(CODASYLモデル)
属性が無く，かつ２つの実体間の関連	I		 （自動，固定）
	II		 （手動，一時）
	III		 （自動，永続）
	IV		 （自動，永続）
	V		 （手動，一時）
	VI		 （自動，固定）
その他の関連	VII		 （自動，固定）

注1) M：1個の実体が M回出現　C：1個の実体が 1または 0回出現　1：1個の実体が 1回出現．
ただし，ケースVII については任意．
2)（　）内は，（挿入属性，切離し属性）を示す．

③親子属性

親子属性は挿入属性（自動，手動）と切離し属性（固定，永続，一時）からなり，６種類ある。関連型に関係付けられる実体型の数および数量関係で表現される実体の結び付き方に着目することにより親子属性を生成する。表２に親子属性の生成パターンを示す。

④親子集合選択

親子集合選択は，子レコードを挿入する親子集合の実現値を決定する方法である。関連型が親子集合に変換されること，関連が実体のキーにより識別されることから，親レコードとなる実体のキーによる親子集合選択を生成する。

⑤親子集合順序

親子集合順序は，親子集合の実現値に子レコードを挿入する位置を決定する方法である。実体関連モデルには対応する概念が無いため，概念スキーマからは生成できない。このため，設計者が追加する必要がある。

図５の概念スキーマに対する論理スキーマ生

```
SCHEMA NAME IS TEST.
 RECORD NAME IS  学校.
   ; RECORD-KEY IS 学校名.
   01  学校名  ; TYPE IS CHARACTER 20.
 RECORD NAME IS  教師.
   ; RECORD-KEY IS 名前.
   01  名前  ; TYPE IS CHARACTER 20.
   01  住所  ; TYPE IS CHARACTER 20.
 RECORD NAME IS  勤務.
   01  赴任日  ; TYPE IS CHARACTER  6.
 :
 SET NAME IS 勤務１
   ; OWNER IS  学校
   ; ORDER IS PERMANENT ***
   MEMBER IS 勤務
   ; INSERTION IS AUTOMATIC RETENTION IS FIXED
   ; SET SELECTION IS THRU 勤務１
         OWNER IDENTIFIED BY KEY  学校名.
 SET NAME IS 勤務２
   ; OWNER IS  教師.
   ; ORDER IS PERMANENT ***
   MEMBER IS 勤務
   ; INSERTION IS AUTOMATIC RETENTION IS FIXED
   ; SET SELECTION IS THRU 勤務２
         OWNER IDENTIFIED BY KEY  名前.
 :
END SCHEMA .
```

図６　論理スキーマの生成例

成例を図６に示す。

５．物理設計

５．１　設計手順

記憶スキーマとして以下を仮定する。

①レコードと記憶レコードの対応は１対１とする

②親子集合は "つなぎ "で実現する

③記憶領域のページの大きさは一定とする

④レコードの配置モードは "CALC" と" VIA SET " の２種類とする

記憶スキーマ生成問題は，目標計画法問題に還元されて解かれている[8]。この方法は，性能上最適な論理スキーマが与えられ，かつ適切な目標値が設定された場合，最適な記憶スキーマを生成できるという利点がある。しかし，検索／更新処理時のアクセス時間（ページ・アクセス回数）等詳細な目標値の設定が必要なこと，論理設計工程のみでは性能上最適な論理スキーマを設計できないことから実用性の面で問題がある。

DBDESIGNでは，大規模なデータベースの設計にも適用できることを目的とした実用的な方法として，次の設計手順を取る。

＜ステップ１＞次節で示す方法に従い，論理スキーマから記憶スキーマを生成する。

この生成法は，大規模なデータベースではアクセス効率の方が重視されることに着目し，アクセス効率に焦点を絞った方法である。

＜ステップ２＞データベースの格納状況，動作状況に関する各種の性能分析データを収集して修正箇所を特定し，アクセス効率，記憶効率の両面からスキーマを最適化する。

５．２　記憶スキーマの生成法

論理スキーマから記憶スキーマを生成する方法を以下に示す。

（１）入力情報

①論理スキーマ，②データ操作言語の系列，③レコード，親子集合のデータの件数，④レコードと記憶領域の対応。

（２）生成法

(i) レコードの配置モードの決定

①②③から，レコードのキーアクセス回数と親子集合アクセス回数を算出し，前者の回数が多ければ"CALC"，少なければ "VIA SET"とする。

(ii)索引付けの要／不要，設定するつなぎの種類の決定

①～④と (i)で決定したレコードの配置モードから，索引付けの有無，つなぎの設定パターン毎の I/O回数を算出し， I/O回数が最も少ないパターンとする。 I/O回数は，オカレンスが均一に配置されていると仮定した簡易式により算出する。

(iii) 記憶領域の必要容量

③④および (i)で決定したレコードの配置モードから算出する。

５．３　性能分析データの収集法[10,11]

性能分析データ収集の入出力情報，収集方式を以下に示す。

（１）入力情報

①論理スキーマおよび記憶スキーマ，②レコード，親子集合のデータの件数，③データ操作言語の系列。

（２）出力情報

マクロな情報，ミクロな情報および動作状況分析データ，格納状況分析データに分類して出力する。主な出力項目を表２に示す。

（３）収集方式

最初に，①，②から，各記憶領域のページ当りのオカレンス数を求め，格納状況分析データを算出する。次に，格納状況分析データに基づいて，各業務プログラムのレコード，親子集合へのアクセスを疑似し，その過程でのアクセス回数を蓄積，編集することにより動作状況分析データを算出する。

表3　主な性能分析データ収集項目

分析データ	分　類	主な収集項目
格納状況分析データ	<マクロな情報>	データベース容量
	<ミクロな情報>	
	レコード	ページ内オカレンス数
	親子集合	セット・オカレンス長
	索引	索引ツリーのレベル数
動作状況分析データ	<マクロな情報>	AP毎の CPU時間，I/O 回数
	<ミクロな情報>	左記の項目対応のアクセス回数，I/O 回数．
	レコード	
	データ項目	
	親子集合	
	索引	

６．おわりに

データベース設計の容易化，効率化を目的とするデータベース設計システムDBDESIGNについて述べた。主な内容は以下の通りである。

（１）実体関連モデルによる概念スキーマの設計手法

（２）概念スキーマからアクセス効率を考慮した CODASYL型論理スキーマを生成する方法

（３）アクセス効率の良い記憶スキーマの生成法

（４）性能上の問題点の特定が可能な，詳細な性能分析データの収集法

今後の課題として以下が考えられる。

（１）概念／論理／格納スキーマの診断機能

（２）関係データベースに対する設計支援

［謝辞］

日頃，ご指導戴く寺島室長，並びにDBDESIGN関係各位に感謝致します。

［参考文献］

[1]V.Y.Lum,et al : 1978 New Or leans Database Design Workshop Report,Proc. 5th VLDB, 1979.

[2]C.Batini,M.Lenzerini : A Methodology for Data Schema Integration in the Entity-Relationship Model, in Entity-Relationship Approch to Software Engineering 1983.

[3]P.P.Chen : The Entity-Relationship Model Toward a Unified View of Data , ACM-TODS,vol.1,No.1,1976.

[4]S.B.Navathe,S.G.Gadqil : A Methodlogy for View Integration in Logical Database Design, Proc. 8th VLDB,1982 .

[5]S.B.Navathe,T.Sashidhar,R.Elmasri : Relationship Merging in Schema Integration, Proc. 10th VLDB,1984.

[6]CODASYL.Report of the CODASYL data definition language committee. Inf.Syst.3,4, 1978.

[7]K.B.Irani : A Designer for DBMS-Processable Logical Database Structure, Proc. 5th VLDB,1979 .

[8]H.K.Jain : Goal Programming Approch to Physical Design of Network Database ,Proc. COMPSAC , 1983.

[9]T.J.Teorey,L.B.Oberlander : Network Database Evaluation Using Analytical Modeling , N.C.C., 1978.

[10]馬場，中野，山多：データベース設計における分析データ収集に関する考察，情処第29回全国大会，1984.
[11]山多，中野，橋本：データベース設計用性能推定の方式について，情処DBMS研究会 38-2，1983.