データベース概念スキーマの設計における情報構造およびトランザクション定義の一手法†

酒井 博敬††

ER (Entity-Relationship) モデルを用いたデータベースの概念スキーマ設計の方法論について論じた．
ER モデルのアプローチによれば，概念スキーマはその初期設計において直観的な手続きで作成される．関係モデルにおける正規化および汎化の概念をERモデルに適用することによって，また実現値化および類型化の概念を新たに導入することによって，初期スキーマを意味論的に自然かつ明快な概念スキーマに改良することができる．
次に，データベースに対する処理要求をトランザクション記述として形式化する．この記述によって更新波及が明示的に表現され，トランザクションの特性の把握を容易にすることができる．またトランザクション記述から，トランザクションがもたらすデータベースに対するアクセス操作の局所性を，アクセス荷重として定量化することができる．

1. まえがき

データベースの論理設計の目標は，物理データベースの抽象化である概念データベースを設計することにある．概念データベースの情報構造は概念スキーマとして定義される．ここではChenによって提案されたER (Entity-Relationship) モデルを用いて概念スキーマを表現する[1),2)]．

ERモデルは情報構造に関する意味表現力にすぐれ，ユーザと設計者の双方がシステムへの要求に対する共通の認識と理解をもつための枠組として好適である．またERモデルから階層，ネットワーク，および関係の各データ・モデルへの変換が容易であることから[1),4)]，ERモデルにおいて展開された設計技法は，現実のデータベース論理設計に直ちに適用することができる．

概念スキーマにはデータベースに対する多様な情報要求が正しく反映されなければならない．そのためには情報要求を構造特性および処理特性の両面から解析し，スキーマの表現能力を高める必要がある．

Chenは実世界における実体と実体間の関係をトップダウン的なアプローチでとらえ，情報構造特性をERモデルに表現した．その後関係モデルにおける正規化 (normalization) および汎化 (generalization) の概念のERモデルへの適用が，それぞれSakai[3),5)]およびScheuermann-Schiffner-Weber[7)]によってなされ，ERモデルによるスキーマ改良手続きの定式化が導入された．これらの概念に加え，新たに実現値化 (instantiation) および類型化 (typification) の概念を導入することによって，概念スキーマに対してより自然かつ明快な意味表現力を与えることができる．この論文では，スキーマ改良手続きの体系化を目的として，ERモデルにおける正規化の適用，Scheuermann等による考えを改善した形での汎化の適用，および実現値化と類型化による改良手続きの定式化について述べる．

データベースは時間とともに変化する．Santos-Neuhold-Furtadoは ERモデルにおける処理特性を存在制約と操作制約の面から考察した[6)]．実用的見地からは，これらの完全性に関する制約はデータベースに対する問合せと更新要求の記述に直接反映されることが望ましい．ここでは情報処理特性の1つの表現手段として，問合せおよび更新要求を形式化したトランザクション記述を導入し，スキーマの表現を補うことを試みた．

2. ERモデルとER図

ERモデルでは，情報は3つの概念要素，すなわち実体 (entity)，関連 (relationship)，および値 (value) によって表現される．同種の実体の集合，同種の関連の集合，および同種の値の集合を，それぞれE集合，R集合，およびV集合という．

E集合はE ($A_1/V_1, A_2/V_2, \cdots, A_n/V_n$) のように記述される．EはE集合名，$A_i/V_i$ ($i=1, 2, \cdots, n$) は属性とV集合の対である．属性 A_i はE集合の性質を表わ

† A Methodology for Defining Information Structures and Transactions in the Conceptual Schema Design by HIROTAKA SAKAI (Hitachi Institute of Technology, Hitachi, Ltd.)

†† (株)日立製作所日立技術研修所

し，Eから V_i への関数として定義される．属性の集合 $X=\{A_1, A_2, \cdots, A_k\}$ がEをV集合の直積 $V_1 \times V_2 \times \cdots \times V_k$ へ1対1に写像する極小集合であるとき，XをEの識別子という．以後E集合の記述は簡略化して，$E(\underline{A_1, A_2, \cdots, A_k}, A_{k+1}, \cdots, A_n)$ のように書く．下線をつけた部分は識別子である．2種以上の識別子がある場合は代表を1つ選んで下線をつけることにする．

R集合はいくつかのE集合 $E_1, E_2, \cdots, E_m$ (必ずしも相異なるものとはかぎらない) を関係づける集合である．すなわち互いに関係づけられる実体 $e_i \in E_i (i=1, 2, \cdots, m)$ の組 $(e_1, e_2, \cdots, e_m)$ を関連といい，同種の関連の集合がR集合である．R集合は $R(E_1/L_1, E_2/L_2, \cdots, E_m/L_m : A_1/V_1, A_2/V_2, \cdots, A_n/V_n)$ のように記述される．RはR集合名，$E_i/L_i (i=1, 2, \cdots, m)$ はE集合と役割の対である．すなわち Li は E_i がR集合Rの中で果す役割を表わす名称である．R集合も属性とV集合の対 $A_j/V_j (j=1, 2, \cdots, n)$ をもつことがある．属性 A_j はRから V_j への関数である．以後R集合の記述は簡略化して，$R(E_1, E_2, \cdots, E_m : A_1, A_2, \cdots, A_n)$ のように書く．

ER モデルはE集合とR集合の集合として構成される．ER モデルの図式表現を ER 図という．ER 図では，E集合を長方形，R集合を菱形で表わし，両者を実線で結んでいくつかのE集合がR集合によって関係づけられていることを表示する．

例 1：AIRLINE データベース (1)

図1のような，E集合 FLIGHT, PASSENGER, FDATE, およびR集合 RESERVE, AVAILABLE からなる概念スキーマを考える．ここで RESERVE は PASSENGER の特定日付け (FDATE) の FLIGHT に対する予約関係を表わし，またR集合AVAILABLE は特定日付けにおける FLIGHT の空席数を知るために設定されている．各集合の記述は次で与えられる．

[E集合]

PASSENGER (NAME, ADDRESS, PHONE).

FLIGHT (FLT#, SOURCE, DEST, DEP-TIME, ARR-TIME, PLANE-TYPE, SEAT-CAPACITY).

FDATE (DATE).

[R集合]

RESERVE (PASSENGER, FLIGHT, FDATE: NO-OF-SEATS).

AVAILABLE (FLIGHT, FDATE: NO-OF-SEATS-AVAILABLE).

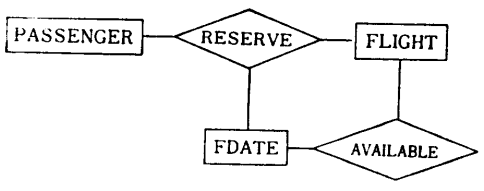

図 1 AIRLINE データベース (1) の ER 図
Fig. 1 ER diagram of the AIRLINE database (1).

3. 概念スキーマの改良手続き

ER モデルによる概念スキーマは，(i) E集合をきめる．(ii) R集合をきめる．(iii) ER 図を作る．(iv) V集合をきめ，E集合とR集合の属性を定義する．という手順で初期設計される．初期設計ではE集合，R集合などのスキーマ要素は直観的に認識される．ここで得られた概念スキーマを，正規化 (normalization), 汎化 (generalization), 実現値化 (instantiation), および類型化 (typification) の観点から見直すことによって，より豊富な内容と明快な意味づけをもつスキーマに改良することができる．

(1) 正 規 化

関係モデルにおける正規化概念を，ER モデルにおいて次のように定義して用いることにする．すなわちどのE集合およびR集合においても，E (あるいはR) の識別子に含まれない属性は互いに関数従属でなく，かつE (あるいはR) の識別子に完全関数従属であるとき，ER モデルは正規化されているという．ここでE (あるいはR) における属性の集合間の関数従属および完全関数従属の概念は，属性に対応するV集合 (またはV集合の直積) 間の関数従属および完全関数従属として定義される．またとくにR集合の正規化を考える場合，Rの識別子として，Rによって関係づけられたE集合の識別子の集合を考えるものとする．ER モデルの正規化について，Chen は直観的な考察を行ったが[1]，その後 Sakai によって，スキーマ変換の形式化と，E集合あるいはR集合としての意味論的解釈を組み合せたスキーマ正規化手順が得られている[3),5)]．

例 2：AIRLINE データベース (2)

例1の AIRLINE データベース (1) におけるE集合 FLIGHT において，属性 SEAT-CAPACITY は識別子 FLT# のほかに属性 PLANE-TYPE にも関数従属である．そこでE集合 FLIGHT を，改めて次のE集合 FLIGHT, PLANE, およびR集合 ASSIGN に分解することによって，**図2**に示される概念スキー

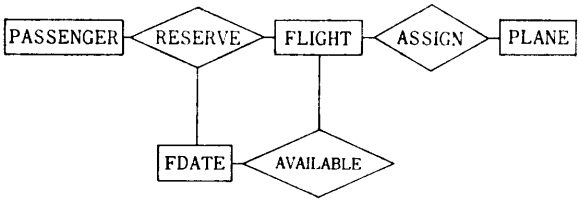

図 2 AIRLINE データベース（2）の ER 図
Fig. 2 ER diagram of the AIRLINE database (2).

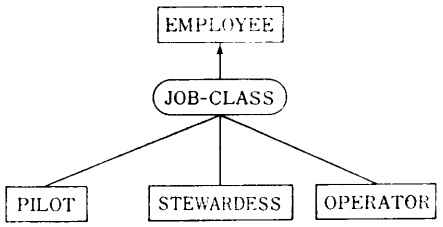

図 3 ER 図による汎化の表現
Fig. 3 ER diagram representing the generalization.

マの形に正規化することができる.

[E集合]

FLIGHT (FLT♯, SOURCE, DEST, DEP-TIME, ARR-TIME).

PLANE (PLANE-TYPE, SEAT-CAPACITY).

[R集合]

ASSIGN (FLIGHT, PLANE).

（2） 汎化と特化

Smith-Smith によって導かれた関係モデルにおける汎化 (generalization) と特化 (specialization) の概念[8]を，次の形で ER モデルに適用する．すなわち汎化とは，あるカテゴリにしたがって類似のE集合を集めて1つのクラスを作り，これを単一のE集合とみなす抽象化をいう．逆に単一のE集合をあるカテゴリにしたがっていくつかのE集合に類別することを特化という．Scheuermann は ER モデルにおける汎化の概念を考察し，カテゴリをクラスという名前でR集合の一種として扱った[7]．しかし汎化の概念は実体や関連などの対象と異なる抽象化レベルに分離する働きをもっており，ER 図においても汎化を含む ER モデルは本来3次元の図形として表現するのが適切である．ここでは ER 図の簡略化のために，**図3**のように独立な概念要素としてのカテゴリを長円形で表わし，長円形から汎化されたE集合への矢線によって汎化を表わすことにする．

例 3：航空会社におけるE集合 EMPLOYEE は，E集合PILOT, STEWARDESS, およびOPERATORをカテゴリ JOB-CLASS によって汎化したものである．これは図3の ER 図に表わされる．ここで特化された各E集合は，上位のE集合 EMPLOYEE の属性を引き継ぐとともに，それぞれ独自の属性をもつ．

（3） 実 現 値 化

2つのE集合EおよびFがR集合Rによって多対多の関係をもつ場合を考える．Eの実体 e_0 に対して，集合 $R/e_0=\{(e_0, f)\mid f\in F, (e_0, f)\in R\}$ を Rにおける e_0 のF別実現値集合 (instance set) とよぶ．R集合RをE集合とみなし，Eの実体 e_0 に対してF別実現値集合 R/e_0 を対応させ，その各要素を e_0 の実現値として扱うことをE集合Eの実現値化ということにする．実現値化の概念は，とくにFが時間要素を表わすE集合の場合，Eの実体 e_0 に対して，e_0 の時系列的な実現値 $(e_0, f_1), (e_0, f_2), \cdots, (e_0, f_k)(f_i\in F)$ を実体として扱うときに有効である．

実現値化においては，R集合RをE集合として定義し直すために，形式的に以下の手続きを必要とする．

（i） Rの要素 r を実体とみなすとき，これを r^* と書くことにし，E集合 $R^*=\{r^*\mid r\in R\}$ を考える．E集合EおよびFの識別子をそれぞれ X および Y，Rの属性を Z とするとき，R*の属性を X^*, Y^*, および Z^* と定める．ただし $r=(e, f)$ とするとき，$X^*(r^*)=X(e)$, $Y^*(r^*)=Y(f)$, および $Z^*(r^*)=Z(r)$ とする．

（ii） E集合EとR*をR集合 INSTANCE-OF-Eによって関係づける．このR集合によって，Eの実体 e_0 は R* におけるF別実現値集合 $\{r^*\mid r\in R/e_0\}$ に対応づけられる．

例 4：AIRLINE データベース（3）

AIRLINE データベース（2）のR集合 AVAILABLE (FLIGHT, FDATE: NO-OF-SEATS-AVAILABLE) において，E集合 FLIGHT を FDATE 別に実現値化する．すなわちR集合 AVAILABLE に代えて，E集合 OPEN-FLIGHT (FLT♯, DATE, NO-OF-SEATS-AVAILABLE) を設け，これをR集合 INSTANCE-OF-FLIGHT によってE集合 FLIGHT に関係づける．E集合 FDATE はE集合 OPEN-FLIGHT の属性 DATE によって表現されるからスキーマから削除する．次にR集合 RESERVE を改めて，E集合 PASSENGER と OPEN-FLIGHT を関係づけるR集合 RESERVE (PASSENGER, OPEN-FLIGHT: NO-OF-SEATS) として定義する．以上の修正を施すことによって，概念スキーマにより

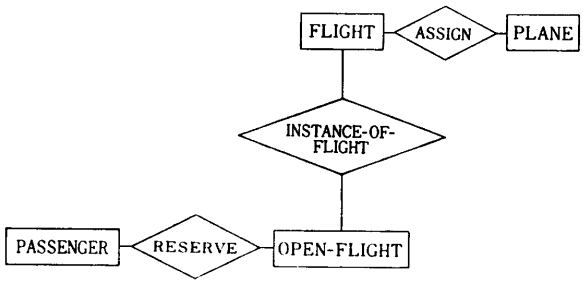

図 4 AIRLINE データベース（3）の ER 図
Fig. 4 ER diagram of the AIRLINE database (3).

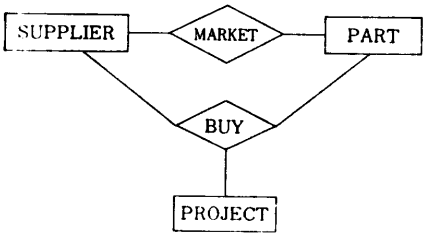

図 5 SUPPLIER-PART-PROJECT（1）の ER 図
Fig. 5 ER diagram SUPPLIER-PART-PROJECT (1).

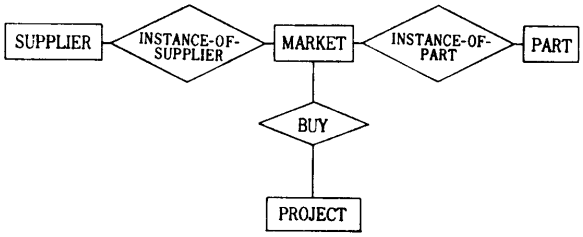

図 6 SUPPLIER-PART-PROJECT（2）の ER 図
Fig. 6 ER diagram SUPPLIER-PART-PROJECT (2).

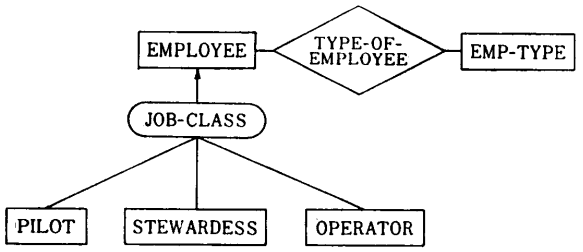

図 7 TYPE-OF R 集合を含む ER 図
Fig. 7 ER diagram representing the TYPE-OF R set.

自然な意味をもたせることができる．図4に実現値化を施した後の ER 図を示す．

例 5：実現値化の考え方は，時間要素を表わすE集合が含まれない場合にも適用できる．たとえば図5のスキーマにおいて，R集合 MARKET はE集合 SUPPLIER および PART 間の販売関係を表わし，R集合 BUY はE集合 PROJECT がどの PART をどの SUPPLIER から購入するかという関係を表わす．BUY の要素が存在するためには，これに関与する SUPPLIER と PART の要素を含む MARKET の要素が存在しなければならない．

ここでR集合 MARKET が SUPPLIER と PART 間の多対多の関係であることに着目し，MARKET において，それぞれ SUPPLIER を PART 別に，また PART を SUPPLIER 別に実現値化する．そしてE集合として再定義された MARKET をR集合 BUY によって PROJECT に関係づけ，図6のスキーマを得る．この修正によって，R集合 BUY の MARKET に対する存在依存性は自明な形で ER 図に表現される．

（4） 類 型 化

E集合Eがある基準にしたがっていくつかのグループに類別されるとき，各グループに対してその型(type)を表わす実体を対応させることをEの類型化とよぶ．ここで類型化の形式的手続きを次のように定めることにする．

（i） E集合Eが互いに素な部分集合 $E_1, E_2, \cdots, E_k$ に類別されるとする．この類別は，特化による類別，実現値化における実現値集合への類別，そのほか任意の基準による類別でよい．

（ii） 各 E_i に対してその型を表わす実体 $\bar{e}_i$ を考え，それから構成されるE集合 $\bar{\mathrm{E}}=\{\bar{e}_i\}$ を定義する．$\bar{e}_i$ は E_i の全要素に共通的な性質あるいは E_i の全要素から導き出される性質をもつ実体である．

（iii） $\bar{\mathrm{E}}$ とEをR集合 TYPE-OF-E によって関係づける．TYPE-OF-E は $\bar{e}_i$ と E_i の全要素を関係づけるR集合である．

例 6：例3のようにE集合 EMPLOYEE が特化によって PILOT, STEWARDESS, および OPERATOR に類別されているとき，これを類型化してE集合 EMP-TYPE を次のように定義する．

EMP-TYPE (TYPE-ID, JOB-DESCRIPTION, NO-OF-EMP, MAX-SALARY, MIN-SALARY, AVERAGE-AGE).

このE集合 EMP-TYPE は，PILOT, STEWARDESS, および OPERATOR の各E集合の型を表わす3つの実体からなる．EMP-TYPE と EMPLOYEE をR集合 TYPE-OF-EMPLOYEE によって関係づけて図7のスキーマを得る．

例 7：AIRLINE データベース（3）において，E集合 OPEN-FLIGHT は同一の FLT # の値をもつグループに類別されるから，これを類型化して次のE集

合 FLT-RECORD を定義することができる.

FLT-RECORD (FLT #, AVERAGE-NO-OF-PASSENGERS, TOTAL-SALES, AVERAGE-COST-OF-FUEL).

一方E集合FLIGHTも,OPEN-FLIGHTのFLT #による類別に対する類型化とみなすことができる. FLT-RECORD と FLIGHT は, OPEN-FLIGHT の各グループの動的な性質と静的な性質について, それぞれ類型化したものと考えてよい. 両者を分離することによって, それぞれのE集合の意味を明快に把握することができる. 一方視点によっては両者を合体することもできる.

4. トランザクション解析

データベースは時間とともに変化する. この変化に関する意味的な規則は, ある種の完全性制約, とくにE集合およびR集合に関する存在制約と操作制約の形で表現される[6]. 実用的見地からは, これらの制約はデータベースに対する問合せおよび更新要求の記述に直接的な形で反映されることが望ましい.

ここでは問合せおよび更新要求をトランザクション記述として形式化し, この記述の中に, アクセス経路, 操作手順, および更新波及を明示的に表現することによって, トランザクションの特性を明快にすることを考える. さらにトランザクションの実行頻度や要求される応答時間をトランザクション記述の枠の中で分析することによって, トランザクションによって引き起されるデータベースに対するアクセス操作の局所性を, 個々のE集合およびR集合にかかるアクセス荷重 (access weight) として定量化することを考える.

以後の例では例4にあげた AIRLINE データベース (3) に対するトランザクションをとりあげる.

4.1 トランザクション記述

データベースへの問合せや更新の要求をトランザクションとよぶ. トランザクション処理のためのデータベース操作において, アクセス経路上に高々1つのR集合が現われるとき, このトランザクションを単純トランザクションとよぶことにする. すなわち単純トランザクションは高々1つのR集合を参照するトランザクションである. 一般のトランザクションTは単純トランザクションの列 $S_1, S_2, \cdots, S_n$ に分解される. これを $T=S_1S_2\cdots S_n$ のように書く.

単純トランザクションは, 以下のように1次形式および2次形式として記述される.

(1) 1次形式 $S=op(U)$.

op は操作, Uは操作の対象を表わす. op は get, cr (create), del (delete), および mod (modify) のいずれかである. 1次形式は op の種類に応じてそれぞれ次の意味を表わす.

(a) $S=get(E)$. EはE集合名である. この1次形式は, Eの属性に関する条件のみからそれを満たすEの部分集合を検索することを表わす.

(b) $S=cr(E)$. EはE集合名である. この1次形式はある実体または実体の集合をEに追加することを表わす.

(a)または(b)の1次形式において, Eから検索される. またはEに追加されるEの部分集合を, 単純トランザクションSの目的空間といい, Sで表わす.

(c) $S=del(S')$ または $S=mod(S')$.

単純トランザクション S′ の目的空間 S′ を削除または変更することを表わす. S′ は後述する2次形式の単純トランザクションの目的空間であってもよい. この場合S′はE集合またはR集合の部分集合である.

例 8: 更新トランザクション T_8 "10月1日の JL 001 便の空席数を0にせよ".

このトランザクションは, OPEN-FLIGHT から所定の条件を満たす実体を検索し, これを更新する単純トランザクションに分解されるから, 次のように記述される.

$T_8=S_{81}S_{82}$.

ただし, $S_{81}=get(\text{OPEN-FLIGHT})$, $S_{82}=mod(S_{81})$.

(2) 2次形式

$S=S_1S_2\cdots S_k(R): op(U)$.

op は操作, Uは操作の対象を表わす. op は get および cr のいずれかである. $S_i(i=1,2,\cdots,k)$ はS以前に実行される単純トランザクション, S_i はその目的空間, またRは参照されるR集合名である. S_i は2次形式であってもよい. この場合も, get あるいは cr によってUから検索される, あるいはUに追加される要素の集合として, S_i の目的空間 S_i が定義される. 2次形式は次の意味を表わす.

(a) $S=S_1S_2\cdots S_k(R): get(U)$.

UはE集合名またはR集合名である. この形式は, $S_1, S_2, \cdots, S_k$ とRによって結合されたE集合U, または $S_1, S_2, \cdots, S_k$ を含むR集合 U=R を対象として, Uの属性に関する与えられた条件を満たすUの部分集合を検索することを表わす.

(b) $S=S_1S_2\cdots S_k(R): cr(R)$.

この形式は，$\underline{S_1}, \underline{S_2}, \cdots, \underline{S_k}$ をR集合Rの意味で結合して関連または関連の集合を作り，これをRに追加することを表わす．

例9：予約確認トランザクション T_b "PASSENGER A氏の10月1日の JL 001 便の予約状況を確認せよ"．

このトランザクションは，PASSENGER と OPEN-FLIGHT からそれぞれ該当する実体を検索し，次に両者を結ぶ関連を RESERVE から検索すればよいから，次のように記述される．

$T_b = S_{b1}S_{b2}S_{b3}$.

ただし，$S_{b1} = \underline{get}$ (PASSENGER), $\underline{S_{b2}} = get$(OPEN-FLIGHT), $S_{b3} = \underline{S_{b1}S_{b2}}$ (RESERVE) : $\underline{get}$ (RESERVE) である．

例10：予約受付トランザクション T_c "PASSENGER B氏の11月3日 の AF 003 便の予約要求を処理せよ"．

このトランザクション T_c は ① OPEN-FLIGHT を検索し，空席数を調べる，②予約可能であればB氏を PASSENGER に登録する．③該当する OPEN-FLIGHT および PASSENGER の実体を結ぶ関連を RESERVE に追加する．④ OPEN-FLIGHT の該当する実体の空席数を更新する，という手順で処理されるから次のように記述される．

$T_c = S_{c1}S_{c2}S_{c3}S_{c4}$.

ただし，$S_{c1} = \underline{get}$ (OPEN-FLIGHT), $S_{c2} = \underline{cr}$ (PASSENGER), $S_{c3} = \underline{S_{c1}S_{c2}}$ (RESERVE) : $\underline{cr}$ (RESERVE), $S_{c4} = \underline{mod}$ $(\underline{S_{c1}})$ である．

例11：予約取消しトランザクション T_d "PASSENGER C氏の予約をすべて解消せよ"．

このトランザクションは，① PASSENGER からC氏を検索する，② OPEN-FLIGHT からC氏が予約したすべての実体を検索する，③ RESERVE から，C氏とC氏が予約した OPEN-FLIGHT の実体を結ぶ関連を検索し，④これを消去する，⑤ OPEN-FLIGHT の座席数を更新する，⑥ PASSENGER からC氏を消去する，という手順で処理される．したがって T_d の記述は次のようになる．

$T_d = S_{d1}S_{d2}S_{d3}S_{d4}S_{d5}S_{d6}$.

ただし，$S_{d1} = \underline{get}$ (PASSENGER), $S_{d2} = \underline{S_{d1}}$ (RESERVE) : $\underline{get}$ (OPEN-FLIGHT), $S_{d3} = \underline{S_{d1}S_{d2}}$ (RESERVE) : $\underline{get}$ (RESERVE), $S_{d4} = \underline{del}$ $(\underline{S_{d3}})$, $S_{d5} = \underline{mod}$ $(\underline{S_{d2}})$, $S_{d6} = \underline{del}$ $(\underline{S_{d1}})$ である．

一般に実体または関連の削除をともなうトランザクションの波及効果は大きい．この例にも示されるように，更新波及効果はトランザクション記述に明示的に表現されている．

概念スキーマ設計の段階においては，各トランザクションの詳細な記述は必要でなく，本節に定義した程度の詳しさで十分である．むしろトランザクション記述が，E集合およびR集合に対する更新波及を明らかにする機能を果たすことに意味がある．

4.2 トランザクションによるアクセス荷重

トランザクションによって引き起されるデータベースに対するアクセス操作の局所性を，E集合およびR集合にかかるアクセス荷重として定量化することができる．この測度を，トランザクション種別およびE集合，R集合別に展開することによって，トランザクションによってもたらされる概念データベースに対するアクセス荷重の分布を把握することができる．

トランザクションTの記述を $T = S_1S_2\cdots S_n$ とする．Tのもたらすアクセス荷重は，3つの要因，すなわちTの複雑さ（ここでは単純トランザクションの個数 n），Tの単位時間当りの発生回数 f，および応答時間に対するきびしさ r（r は要求応答時間が短いほど大きい値をとる量）に依存すると考え，Tのもたらすアクセス荷重 L(T) を nrf と定義する．L(T) は単純トランザクション $S_i (i=1, 2, \cdots, n)$ に平等に配分されると考えることができる．S_i の操作対象は E 集合，R 集合，あるいはそれらの部分集合であるから，次のようにして単純トランザクション S_i によってE集合およびR集合に加えられるアクセス荷重を定義する．

（a） S_i が1次形式 $S_i = op(U)$ （UはE集合またはR集合）の場合，Uに対する S_i によるアクセス荷重は rf であると考え，これを $w(U, S_i)$ で表わす．

（b） S_i が2次形式 $S_i = \underline{S_{i1}S_{i2}\cdots S_{ik}}$ (R) : op (U) （UはE集合またはR集合）の場合，R参照のもとでのUに対する S_i によるアクセス荷重は rf であると考え，これを $w(U/R, S_i)$ で表わす．

（c） Uを対象とするすべての1次形式 S_i についての $w(U, S_i)$ の総和を，Uに対する総アクセス荷重といい，W(U) で表わす．同様にR参照のもとでUを対象とするすべての2次形式 S_i についての $w(U/R, S_i)$ の総和を，R参照のもとでのUに対する総アクセス荷重といい，W(U/R) で表わす．

W(U) および W(U/R) によって，トランザクションのもたらす概念データベースに対するアクセス荷重の分布を知ることができる．

	PASSENGER	OPEN-FLIGHT	OPEN-FLIGHT/ RESERVE	RESERVE/ RESERVE
T_b (予約確認)	S_{b_1}: 30	S_{b_2}: 30		S_{b_3}: 30
T_c (予約受付)	S_{c_3}: 50	S_{c_1} & S_{c_4}: 100		S_{c_3}: 50
T_d (予約取消)	S_{d_1} & S_{d_6}: 10		S_{d_2} & S_{d_5}: 10	S_{d_3} & S_{d_4}: 10
W (総負荷)	90	130	10	90

図 8 AIRLINE データベース (3) に対するトランザクションによるアクセス荷重分布

Fig. 8 Distribution of the access weights imposed on the AIRLINE database (3) by transactions.

例 12: 先の例 9〜例 11 にあげたトランザクション T_b, T_c, および T_d が AIRLINE データベース (3) に与えるアクセス荷重の分布を調べる. この 3 種のトランザクションにはいずれも同程度の応答時間が要求されるものとし, 3 種とも $r=1$ とする. また各トランザクションの発生頻度をそれぞれ 30, 50, および 5 (単位は k 件/日) とする. T_b, T_c, および T_d はそれぞれ 3 個, 4 個, および 6 個の単純トランザクションに分解されたから, 定義によりトランザクションによるアクセス荷重はそれぞれ $L(T_b)=90$, $L(T_c)=200$, および $L(T_d)=30$ である. 3 種のトランザクションが AIRLINE データベース (3) の E 集合および R 集合に与えるアクセス荷重は図 8 のようになる. 図 8 の下段に示されるように, 各 E 集合および R 集合にかかる総アクセス荷重は, それぞれ W (PASSENGER) =90, W (OPEN-FLIGHT)=130, W (OPEN-FLIGHT/RESERVE)=10, および W (RESERVE/RESERVE)=90 となる.

5. むすび

ER モデルを用いた概念スキーマ設計の方法論として, 次の手続きについて述べた.

(1) 正規化, 汎化, 実現値化, および類型化の概念にもとづくスキーマの改良.

(2) 存在制約, 操作制約をアクセス経路, 操作手順, および更新波及の形で反映するトランザクション記述の形式化, およびこれにもとづくトランザクションによるアクセス荷重分布の把握.

これらの手続きは実用的に意味のある論理設計技法として利用できる. さらにトランザクションによるアクセス荷重はデータベースに対する処理要求を定量的に反映するため, 概念スキーマを ER モデルからほかのデータ・モデルに変換する際, 有用な尺度となる[4),9)].

概念スキーマの設計においては, 情報構造の特性と完全性制約を反映するトランザクションの特性を 1 つのスキーマに統合して記述することが望ましい. ここで述べた手続きは, 意味論的により豊富な内容を表現するスキーマ記述言語と, 完全性制約をより体系的に表現する方法論へ発展させる必要がある.

参考文献

1) Chen, P. P.: The Entity-Relationship Model: Towards a Unified View of Data, Transactions on Database Systems Vol. 1, No. 1, pp. 9-36 (1976).
2) Chen, P. P.: The Entity-Relationship Model: A Basis for the Enterprise View of Data, AFIPS Conf. Proc. Vol. 46, AFIPS Press, pp. 77-84 (1977).
3) Sakai, H.: On the Optimization of the Entity-Relationship Model, 3rd USA-JAPAN Computer Conf. Proc. pp. 145-149 (1978).
4) Sakai, H.: A Unified Approach to the Logical Design of a Hierarchical Data Model, in: Entity-Relationship Approach to Systems Analysis and Design, pp. 61-74 (North-Holland, Amsterdam, 1980).
5) Sakai, H.: Entity-Relationship Approach to the Conceptual Schema Design, Proc. ACM-SIGMOD 1980, Santa Monica, California, pp. 1-8 (1980).
6) Santos, C. S. dos, Neuhold, E. J. and Furtado, A. L.: A Data Type Approach to the Entity-Relationship Model, in: Entity-Relationship Approach to Systems Analysis and Design, pp. 103-119 (North-Holland, Amsterdam, 1980).
7) Scheuermann, P., Schiffner, G. and Weber, H.: Abstraction Capabilities and Invariant Properties Modeling within the Entity-Relationship Approach, in: Entity-Relationship Approach to Systems Analysis and Design, pp. 121-140 (North-Holland, Amsterdam, 1980).
8) Smith, J. M. and Smith, D. C. P.: Database Abstractions: Aggregation and Generalization, Transactions on Database Systems Vol. 2, No. 2, pp. 105-133 (1977).
9) Teorey, T. J. and Fry, J. P.: The Logical Record Access Approach to Database Design, ACM Computing Surveys Vol. 12, No. 2, pp. 179-211 (1980).

(昭和 55 年 11 月 12 日受付)
(昭和 56 年 4 月 27 日採録)