

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第3628030号
(P3628030)

(45) 発行日 平成17年3月9日(2005.3.9)

(24) 登録日 平成16年12月17日(2004.12.17)

(51) Int. Cl.⁷

G06F 12/00

F I

G06F 12/00 513A
G06F 12/00 513J
G06F 12/00 505
G06F 12/00 512

請求項の数 2 (全 28 頁)

<p>(21) 出願番号 特願平4-293587 (22) 出願日 平成4年10月30日(1992.10.30) (65) 公開番号 特開平6-149634 (43) 公開日 平成6年5月31日(1994.5.31) 審査請求日 平成11年10月28日(1999.10.28)</p>	<p>(73) 特許権者 000003078 株式会社東芝 東京都港区芝浦一丁目1番1号 (74) 代理人 100081961 弁理士 木内 光春 (72) 発明者 藤原 睦 神奈川県川崎市幸区柳町70番地 株式会 社東芝 柳町工場内 (72) 発明者 上村 秀一 神奈川県川崎市幸区柳町70番地 株式会 社東芝 柳町工場内 (72) 発明者 米田 清 神奈川県川崎市幸区柳町70番地 株式会 社東芝 柳町工場内</p>
--	--

最終頁に続く

(54) 【発明の名称】 データベース装置

(57) 【特許請求の範囲】

【請求項1】

文字列、整数、および浮動小数点数の中から選ばれた単一種類または複数種類の値を示すワードを、序列をもって複数個並べることにより構成されたラインが、複数の表に登録されたデータベース装置において、

個々の表について、入力されるラインにおける複数の序列位置の中から選択された判定用の位置のワードとして特定値を持つラインを登録するというワードの条件と、前記複数の序列位置の中から前記判定用の位置とは別に選択されたソート用の位置のワードの値でソートするというソート規則とを含む表情報により定義される複数の表の登録条件を管理する管理手段と、

前記表の登録条件に基づき、入力されたラインを登録する複数の表を指定する表指定手段と、

前記表に登録されたラインを、各表に設定されたソート規則によって特定される序列位置のワードの値に基づいて各表の上でソートするソート手段と、

前記表指定手段によって指定された各表の上で、前記ソート手段によってソートされた順序に基づき、ラインをそれぞれ検索する検索手段、

を備えたことを特徴とするデータベース装置。

【請求項2】

3つのワードから構成されるラインを登録するための前記表の登録条件は、

前記ラインを構成する3つのワードの3つの序列位置に対して、この3つの序列位置の

中から選択された1つの位置を第1の位置と定義し、残りの2つの位置を第2、第3の位置と定義した場合に、

前記3つのワードから構成される1つのラインを、

前記第1の位置が前記判定用の位置であって、この位置のワードの値が当該ラインの対応する位置のワードと同じ値を持ち、かつ、前記第2の位置が前記ソート用の位置である第1の表と、

前記第1の位置が前記判定用の位置であって、この位置のワードの値が当該ラインの対応する位置のワードと同じ値を持ち、かつ、前記第3の位置が前記ソート用の位置である第2の表と、

前記第2の位置が前記判定用の位置であって、この位置のワードの値が当該ラインの対応する位置のワードと同じ値を持ち、かつ、前記第1の位置が前記ソート用の位置である第3の表と、

前記第3の位置が前記判定用の位置であって、この位置のワードの値が当該ラインの対応する位置のワードと同じ値を持ち、かつ、前記第1の位置が前記ソート用の位置である第4の表、

という4つの表にそれぞれ登録することを規定する、
ことを特徴とする請求項1に記載のデータベース装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は、データベース装置に関する。

【0002】

【従来の技術】

データベース（DB）技術の目的は、データの内部構造を、それを利用する応用プログラムから独立させることである。

【0003】

従来のDBシステム概念上の枠組みには、階層、ネットワーク、リレーショナルなどの代表的なものがある。それらのうち、リレーショナル・データベース（RDB）が最も理論的に整理されており、現在では標準的な技術となっている。このRDBの基礎概念は、「関係」である。この点について、以下に説明する。

【0004】

例えば、 n 個の集合 S_1, \dots, S_n の直積集合 R を次のように定義する。

【0005】

【数1】

$$R = \{ (s_1, \dots, s_n) \mid s_i \in S_i \text{ for } i = 1, \dots, n \}$$

… 式(1)

このとき、 R の任意の部分集合を S_1, \dots, S_n の間の n 項関係（または単に関係、ないしはリレーション）と呼ぶ。たとえば、2個の集合 S_1, S_2 を次の式(2) 40

【0006】

【数2】

$$S_1 = \{ \text{東京, 神奈川} \}$$

$$S_2 = \{ \text{三鷹, 武蔵野, 川崎, 横浜} \}$$

… 式(2)

このとき、2個の集合 S_1, S_2 の直積集合 $S_1 \times S_2$ は、次の式(3)のように表される。

10

20

30

40

50

【0007】

【数3】

$$S_1 \times S_2 =$$

$$\{ (東京, 三鷹), (東京, 武蔵野),$$

$$(東京, 川崎), (東京, 横浜),$$

$$(神奈川, 三鷹), (神奈川, 武蔵野),$$

$$(神奈川, 川崎), (神奈川, 横浜) \}$$

…式(3)

10

このうち、次の式(4)に示すような部分集合Rが、都県 - 市リレーションである。

【0008】

【数4】

$$R = \{ (東京, 三鷹), (東京, 武蔵野),$$

$$(神奈川, 川崎), (神奈川, 横浜) \}$$

…式(4)

RDBは、このようなリレーションの集合をデータとするDBである。要するに、リレーションは図52、図53に示すような表である。一般的に、表を横に見て1つの行をタプル(組、tuple、レコード)、縦にみて1つの列を属性(attribute)と呼ぶ。図52の表で各属性に付けられた名前を属性名(attribute name)といい、各属性の値を属性値(attribute value、フィールド値)という。また、表の名前をリレーション名(関係名、relation name)という。

20

【0009】

このように、RDBにおいては、リレーションという表を概念的な仲立ちとして、データレコードを計算機に收容する。もちろん、計算機の内部では複雑なデータ構造やアクセス手続きをとっている可能性が高い。しかし、ユーザからはその複雑な実装が見えない。実装に関する情報は隠蔽されている。単なる表、すなわちデータの容器があって、ユーザはその容器にレコードを入れるという利用法である。

30

【0010】

RDBは、次のような4つの処理機能を持つ。:

- 「1.レコードを收容する表を設定する。」
- 「2.表にレコードを登録する。」
- 「3.表からレコードを削除する。」
- 「4.表の中のレコードを検索する。」

そして、RDBの実装は、属性と属性値を指定すれば、レコードの登録処理、削除処理、および検索処理が能率良くできるようになっている。

40

【0011】

RDBの特徴の1つは、表とレコードとの独立性が高いことである。ユーザは、まず表を設定し、しかる後にその表にレコードを登録する。あるレコードをある表に登録しておくか否かはユーザの任意である。検索したレコードのある属性値を変更する場合でも、レコードと表との対応関係に変化はない。ユーザがレコードと表との対応を変更するためには、上記のようなレコード登録、削除の操作が必要である。複数の表を指定してそれらに登録されているレコード間で演算を行って生じたレコードを、指定した新たな表に登録する場合でも、新たに生成されたレコードをどの表に登録するかは、ユーザが管理する。すなわち、RDBにおいて、表とレコードの対応関係は、ユーザが任意の表に任意のレコードを登録するという行為を通してのみ与えられる。

50

【 0 0 1 2 】

先に説明したように、関係は1つの集合である。一般に、集合の定義には、外延的な方法と内包的な方法がある。外延的な方法では、要素を全て列挙することによって集合を定義する。たとえば、0以上9以下の整数全体の集合 S を、次の式(5)に示すように定義するのが外延的な定義である。

【 0 0 1 3 】

【数5】

 $S =$ $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

10

…式(5)これに対して、その集合に属する要素が満たすべき性質を規定するのが内包的な定義である。たとえば、上の集合 S を、次の式(6)のように定義するのが内包的な定義である。ここで、 Z は整数の集合である。

【 0 0 1 4 】

【数6】

 $S = \{n \in Z \mid 0 \leq n \leq 9\}$

… 式(6)

これに対して、表にレコードを登録する行為は、「このレコードはこのリレーション(表)に属し、このレコードが登録されない他のリレーション(表)には属さない。」という 20
ことを、属性と属性値を列挙することによって定義している。これを全てのレコードについて行うのであるから、リレーションは外延的に定義されている。

【 0 0 1 5 】

この結果、RDBは次のような特徴を持つ。まず、リレーションに従って行う処理は、概念的に単純であり、能率良く行える。すなわちプログラムは簡単ですみ、計算量も少ない。たとえば図53において、

「(A) 生年が '63年の者を検索する。」

という処理は、リレーションに従う処理であり、能率良く行うことができる。

【 0 0 1 6 】

【発明が解決しようとする課題】

30

これに対して、RDBにおいて、リレーションに従わない処理は、もともと想定されていなかった処理であるから、能率が悪い。たとえば、図53において、

「(B) 生年が若い順に3番めの者を検索する。」

という処理は、リレーションに従わない処理である。この処理は、仮に図53の表中に「生年の順番」という属性がもともとありさえすれば、容易に実行できるものである。それにもかかわらず、図53の表中においては、「生年の順番」という属性が、予め考えられてはいないので、(B)の処理を行うためには、登録されたレコードのうち生年を有するレコードを全て調べた上で、これらのレコードの順番を定めなければならない。このような検索を実行する場合には、プログラムが長くなり、計算量も多くなる。

【 0 0 1 7 】

40

さらに言及すれば、この場合、(B)の処理で用いる属性「生年の順番」は、(A)の処理で用いる属性「生年」から自然に定まるものである。すなわち、属性「生年の順番」は、属性「生年」を有するレコードの関係の部分集合を規定する内包的な性質として、自然に存在するものである。それにも関わらず、属性「生年の順番」は、このRDBには存在しない属性であるために、この属性を利用した(B)の処理は、容易に実行することができないのである。

【 0 0 1 8 】

以上のことから、RDBは、その設計時に想定された使い方には適合している反面、想定から少し外れた使い方をしようとする、能率が悪いことがわかる。ここで、設計時に想定された使い方とは、外延的な定義に用いた属性と、その値とでレコードを扱うことであ 50

る。また、想定から少し外れた使い方とは、前述の(B)の処理のように、RDBに存在する属性とその値から導かれる性質によって内包的に規定されるレコードを扱うことである。

【0019】

一般的に、DBを設計してから時間がたつと、使い方が設計時の想定から外れて来ることは避けられない。なぜなら、DBの方が、それを利用する応用プログラムよりも寿命が長いからである。つまり、ほとんどの場合、DBは既に存在しており、プログラムは、DBに存在する情報をなんとか利用しようという意図をもって書かれるのである。このようにDBの使い方が設計時の想定から外れる場合として、図53の表に対する前述の処理に関連付けるならば、DBの設計時には、「生年」という属性を検索できるようにしておけば十分だと考えられていたにも関わらず、実際には年齢順の方が使いやすいことが後で判明したりする場合などが挙げられる。このような場合、RDBでは、新たに「年齢順」という属性を設けることになる。これは表の作り直しに当り、レコード数が多いときは大変な作業である。

10

【0020】

このことは、従来のデータベースにおける、表とレコードの対応関係(登録操作)が任意であるという本質的な特徴に起因する。すなわち、ユーザはデータベースの構築、利用にあたって、

「A.レコードの作成、変更」

「B.レコードの表への登録、削除」

20

という2つの操作を関連付けて実行する必要がある。なぜなら、データを保存する際には、データベース中にあるレコードが存在している場合、このレコードの存在だけでなく、そのレコードがどの表に登録されており、どの表に登録されていないかということが重要だからである。したがって、1つのレコードが複数の表に登録されるべき内容を表している場合には、ユーザは、それを判断して、そのレコードを対応する複数の表に対して個別に登録しなければならない。

【0021】

例えば、イベントスケジュール管理のデータベースにおいて、将来のイベントを管理している表(表名: FUTURE EVENT)には、「時刻 イベント」のレコードが、イベントとその発生時刻を表すために登録される。ここで、各イベントには、不定人数の参加者が予定されているとする。これに対して、この予定される参加者に関する表(表名: ATTENDANT EVENT)を作成し、この表に、「参加者 イベント」のレコードを登録する。そして、表「FUTURE EVENT」に登録されている「時刻 イベント」のレコードは、イベントが生起すると削除され、すでに生起したイベントを管理している表(表名: PAST EVENT)に登録される。

30

【0022】

この場合、それぞれの表は、その目的に適した形式のレコードのみを効率的に登録しているが、1つのイベントに関してどのような情報が登録されているかという検索は、3つの表(FUTURE EVENT、ATTENDANT EVENT、PAST EVENT)を個別に調べなければ実行できない。例えば、あるイベントに関する「時刻 イベント」のレコードは、表「FUTURE EVENT」か表「PAST EVENT」かどちらか一方に必ず含まれているはずであるが、これを確認するためには、両方の表を同時に調べなければならない。また、イベント生起時には、一方の表からそのイベントに関するレコードを削除し、かつ、他方の表にそのイベントに関するレコードを登録するという2つの処理を、ユーザの責任において同時に行わなければならない。ここに誤りが生じる可能性があり、かつ、誤りの検出にも手間がかかる。

40

【0023】

このような問題を回避する一つの方法として、「FUTURE EVENT」と「PAST EVENT」という2つの表の別をやめて、例えば、1つの表(表名: EVENT)を作成し、この表「EVENT」に、「FUTURE/PAST 時刻 イベント」

50

というレコードとして、イベントのスケジュールを登録することが考えられる。ここで、フィールド「FUTURE/PAST」は、FUTURE or PASTを意味する。この方法を採用した場合、イベント生起時には、前述の方法のように、異なる表に対する2つの処理（削除と登録）を行う必要なく、「FUTURE/PAST」のフィールド値の変更という1つの処理を行うだけでよいため、誤りが発生する可能性が極めて少なくなり、かつ、誤りの検出も比較的容易になる。

【0024】

その反面、この方法での処理を進めるためには、結局、関連性のあるレコードを登録する表をすべて融合して1つの表とすることになるため、有効に使われない多くのフィールドを含む長尺なレコードを登録することになる。このような無駄の多い長尺なレコードを多数登録することは、レコードを収納する表を設定することによるデータベースの効率的運用を放棄することに他ならない。

10

【0025】

以上のように、表を設定し、その表に対してレコードを任意に登録、削除することでデータを効率的に蓄積利用しようとする従来のデータベースシステムにおいては、表とレコードの対応関係の意図的な管理に多くの労力を要し、その分だけ多くのコストが費やされている。また、従来のデータベースにおいては、レコードの内容すなわちフィールド値に関して、あるフィールド値を持つレコードの指定した表への（正しいレコードとしての）登録の可否を制限することはあっても、そのレコードがどの表に登録されるべきであるかを指示する機能、あるいは、実際に自動的に登録を行う機能は、設けられていない。

20

【0026】

したがって、従来のデータベースにおいて、特定のフィールド値に関して、どのようなレコードがそのフィールド値を有し、また、そのフィールド値を有するレコードがどのような表に登録されているかを検索することは困難である。すなわち、フィールド値は、文字列や数値であり、それが各レコード（の各フィールド）でどのような意味に用いられるかはユーザの任意である。そのため、異なる表の異なるレコード間で、同一のフィールド値が全く異なる意味で用いられることは十分ありうる。このような同一のフィールド値の異なる意味での使用は、データベース中におけるデータ処理の混乱を来し、極めて不都合である。したがって、ユーザがレコードを作成する際には、それぞれのフィールド値が、そのときすでにデータベース中に使用されているフィールド値とどういう関連を持つかを把握することが不可欠である。そして、このように、特定のフィールド値の情報をデータベース全体にわたって完全に把握するためには、従来のデータベースにおいて、関連する（レコードを含む）可能性のある表のレコードをすべて調べて結論を出さなければならない。しかしながら、このような検索の実行には、大変な手間がかかり、また、個々のフィールド値ごとに可能性のある表やフィールドの情報をユーザが維持管理すること自体が困難である。

30

【0027】

本発明は上記のような従来技術の課題を解決するために提案されたものであり、その目的は、新たに登録するデータまたは内容を変更するデータとすでに登録されているデータとの関連を容易に把握可能であり、表とデータの管理が簡略で、設計時に想定されなかった使い方に対しても柔軟に対処でき、操作性および能率に優れたデータベース装置を提供することである。

40

【0028】

【課題を解決するための手段】

本発明によるデータベース装置は、文字列、整数、および浮動小数点数の中から選ばれた単一種類または複数種類の値を示すワードを、序列をもって複数個並べることにより構成されたラインが、複数の表に登録されたデータベース装置において、管理手段、表指定手段、ソート手段、検索手段、を備えたことを特徴とするものである。ここで、管理手段は、個々の表について、入力されるラインにおける複数の序列位置の中から選択された判定用の位置のワードとして特定値を持つラインを登録するというワードの条件と、複数の序

50

列位置の中から判定用の位置とは別に選択されたソート用の位置のワードの値でソートするというソート規則とを含む表情報により定義される複数の表の登録条件を管理する手段である。また、表指定手段は、表の登録条件に基づき、入力されたラインを登録する複数の表を指定する手段である。また、ソート手段は、表に登録されたラインを、各表に設定されたソート規則によって特定される序列位置のワードの値に基づいて各表の上でソートする手段である。さらに、検索手段は、表指定手段によって指定された各表の上で、ソート手段によってソートされた順序に基づき、ラインをそれぞれ検索する手段である。

【0029】

【作用】

以上のような構成を有する本発明の作用は次の通りである。まず、ラインにおける複数の序列位置について、判定用の位置におけるワードの値に応じて当該ラインの登録を判定するためのワードの条件と、ソート用の位置におけるワードの値でソートするためのソート規則とを含む表情報により登録条件を定義しているため、この登録条件により、1つのラインと、このラインを登録すべき複数の表との対応関係を明確に示すことができる。その結果、ラインを登録する際には、表指定手段により、ラインと表の対応関係を示す登録条件に基づき、ラインを構成する複数のワードの値に応じて、収納すべき複数の表を自動的に指定できるため、1つのラインを異なるソート規則を持つ複数の表に容易に登録できる。したがって、設計時に想定されなかった使い方に柔軟に対処可能であり、操作性および能率を向上できる。

【0030】

一方、本発明では、表指定手段により、1つのワードが収納されている表を自動的に特定可能であり、また、表指定手段に加えて検索手段を使用することにより、1つのワードによるラインの検索を容易に行うことができる。したがって、1つのワードの情報をデータベース全体にわたって容易に把握でき、データベースにすでに登録されたラインと新しく登録されるラインとの関連を容易に把握することができる。加えて、ソート手段を利用していることから、ソート順によるラインの検索や、検索されたラインの位置の決定なども自動的に簡単に行えるため、検索処理や、検索の必要な削除処理、変更処理などの能率を向上できる。

【0031】

【実施例】

[装置構成]

以下には、本発明によるデータベース装置の構成について、図1を参照して説明する。ここで、図1は、本発明によるデータベース装置の基本的な機能構成の一例を示す機能ブロック図である。この図1に示す装置は、入力手段1、格納手段2、表指定手段3、管理手段4、登録手段5、ソート手段6、検索手段7、削除手段8、出力手段9を備えており、複数のワードの順序列をラインとして、複数のラインを複数の条件に合致する複数の表に対して個別・選択的に登録・削除するとともに、ワードの内容に基づいてラインを検索するように構成されている。

【0032】

入力手段1は、複数のワードの順序列をラインとして入力する機能と、すでに登録済みのラインを、ワードの内容またはラインの位置によって指定し、検索指令、削除指令、または変更指令を入力する機能を有する。格納手段2は、各表に設定されたソート規則と各表に登録されたライン情報およびそのソート順情報を含む複数の表情報を格納する機能を有する。

【0033】

表指定手段3は、ラインと表の対応関係を示す予め設定された一定の登録条件を有し、入力手段1によるラインの入力に回答して、入力されたラインのワードの内容と登録条件に基づき、そのラインを登録する複数の表を指定する機能と、入力手段1による登録済みのラインの検索指令、削除指令、または変更指令の入力に回答して、指定されたラインのワードの内容と登録条件に基づき、そのラインが登録された複数の表を指定する機能を有す

10

20

30

40

50

る。

【0034】

管理手段4は、格納手段2に格納された複数の表情報を管理し、表指定手段3によって指定された複数の表情報を提供する機能と、登録手段5およびソート手段6による登録情報と、削除手段8による削除情報によって管理情報を更新する機能と、検索手段7による検索情報を管理する機能を有する。

【0035】

登録手段5は、入力手段1によるラインの入力に応答して、表指定手段3によって指定された複数の表情報を管理手段4から受け取り、各表に、入力手段1によって入力されたラインをそれぞれ登録する機能を有する。ソート手段6は、登録手段5によって複数の表に登録されたラインを、各表に設定されたソート規則によって特定される位置のワードの内容に基づいて表の上にソートする機能を有する。

10

【0036】

検索手段7は、入力手段1による登録済みのラインの検索指令、削除指令、または変更指令の入力に応答して、表指定手段3によって指定された複数の表情報を管理手段4から受け取り、各表の上で、ソート手段6によってソートされた順序に基づき、入力手段1によって指定された登録済みのラインをそれぞれ検索する機能を有する。削除手段8は、入力手段1による登録済みのラインの削除指令または変更指令の入力に応答して、検索手段7によって検索された複数の表情報を受け取り、各表の上で検索されたラインを削除する機能を有する。出力手段9は、管理手段4による管理情報を出力する機能を有する。

20

【0037】

また、一般的には、管理手段4は表作成手段10を有している。この表作成手段10は、表指定手段3によって指定された表が格納手段内に存在していない場合に、指定された表に合致する新たな表を作成する機能を有する。一方、ラインを構成するワードとしては、例えば、2つの文字列の順序対、文字列と整数の順序対、2つの整数の順序対、または、1つの浮動小数点数の4種類のいずれかを使用することが可能である。より一般的には、文字列、整数、浮動小数点数、およびそれらの任意個数の順序組を使用することが可能である。さらに、本発明においては、表上のラインをソートするソート手段6として、例えば、二分探索木を使用することが可能であるが、二分探索木と同様に、ラインのソート、検索、登録、および削除が可能であれば、別のソート手段を使用することも可能である。

30

【0038】

次に、本発明において、表指定手段3に予め設定される登録条件、すなわち、ラインと表の対応関係を示す一定の登録条件について説明する。まず、1つのラインは、 n ワード (w_1, \dots, w_n) の順序列で構成される。そして、1つのラインを登録する k 種類の表は、 H_1, \dots, H_k と表される。なお、本発明においては、1つのラインが、必ず k 種類の表に登録されるわけではなく、ラインの内容によって登録される表の数が異なる場合もある。

【0039】

ここで、 $x =$ ワード or ライン、 $N = 1 \sim n$ の数、 $S =$ ソート規則として、各表を (x, N, S) で表すことを考える。この場合、 x がワードであれば、 (x, N, S) は、左から N 番目のワードがワード x であるようなラインからなり、ソート規則が S である表を表す。また、 x がラインであれば、 (x, N, S) は、左から N 番目のワードがライン x の N 番目のワードであるようなラインからなり、ソート規則が S である表を表す。さらに、 $i = 1, \dots, k$ とした場合、 i 番目の表 H_i は (x, N_i, S_i) と表される。

40

【0040】

[処理手順の概要]

図2～図5は、図1に示すデータベース装置の一般的な処理手順の例を示すフローチャートであり、図2は登録処理、図3は検索処理、図4は削除処理、図5は変更処理をそれぞれ示している。以下には、各処理手順の概要について順次説明する。

【0041】

50

まず、ラインを登録する場合には、図 2 に示すように、ステップ 1 において、入力手段 1 により、ラインを構成する n 個のワード (w_1, \dots, w_n) が作成され、ステップ 2 において、 n 個のワード (w_1, \dots, w_n) の順序列であるライン L が作成され、入力される。ステップ 3 において、表指定手段 3 により、ライン L のワードの内容と予め設定された登録条件に基づき、そのライン L を登録する k 種類の表 H_1, \dots, H_k が自動的に指定される。ステップ 4 において、管理手段 4 により、指定された複数の表が格納手段 2 内に存在するか否かが判断され、指定された表 H_1, \dots, H_k のうちで存在しない表がある場合には、ステップ 5 において、表作成手段 10 により、必要な表が作成される。ステップ 6 において、指定された k 種類の表 H_1, \dots, H_k の情報が登録手段 5 に提供され、登録手段 5 により、各表にライン L が登録される。ステップ 7 において、ソート手段 6 により、各表に設定されたソート規則 S に基づいて各表がソートされ、登録処理を終了する。

10

【0042】

次に、ラインを検索する場合には、図 3 に示すように、ステップ 11 において、入力手段 1 により、検索するライン L が指定され、ステップ 12 において、表指定手段 3 により、ライン L のワードの内容と予め設定された登録条件に基づき、ライン L が登録された k 種類の表 H_1, \dots, H_k が自動的に指定される。ステップ 13 において、指定された k 種類の表 H_1, \dots, H_k の情報が検索手段 7 に提供され、この検索手段 7 により、指定されたライン L が各表においてソート順に検索され、検索処理を終了する。

【0043】

また、ラインを削除する場合には、図 4 に示すように、ステップ 21 において、入力手段 1 により、削除するライン L が指定され、ステップ 22 において、表指定手段 3 により、ライン L のワードの内容と予め設定された登録条件に基づき、ライン L が登録された k 種類の表 H_1, \dots, H_k が自動的に指定される。ステップ 23 において、指定された k 種類の表 H_1, \dots, H_k の情報が検索手段 7 に提供され、この検索手段 7 により、指定されたライン L が各表においてソート順に検索される。ステップ 24 において、指定された各表と各表における検索されたライン位置の情報が削除手段 8 に提供され、削除手段 8 により、各表からライン L が削除され、削除処理を終了する。

20

【0044】

一方、ラインを変更する場合には、図 5 に示すように、ステップ 31 において、入力手段 1 により、変更するライン L が指定され、ステップ 32 およびステップ 33 において、変更後の n 個のワードとその順序列である変更後のライン L' が作成され、入力される。ステップ 34 において、表指定手段 3 により、変更前のライン L のワードの内容と予め設定された登録条件に基づき、変更前のライン L が登録された k 種類の表 H_1, \dots, H_k が自動的に指定される。ステップ 35 において、指定された k 種類の表 H_1, \dots, H_k の情報が検索手段 7 に提供され、指定されたライン L が各表においてソート順に検索される。ステップ 36 において、指定された各表と各表における検索されたライン位置の情報が削除手段 8 に提供され、削除手段 8 により、各表から変更前のライン L が削除される。ステップ 37 において、表指定手段 3 により、変更後のライン L' のワードの内容と予め設定された登録条件に基づき、そのライン L' を登録する k 種類の表 H_1, \dots, H_k が自動的に指定される。ステップ 38 において、管理手段 4 により、指定された複数の表が格納手段 2 内に存在するか否かが判断され、指定された表 H_1, \dots, H_k のうちで存在しない表がある場合には、ステップ 39 において、表作成手段 10 により、必要な表が作成される。ステップ 40 において、指定された k 種類の表 H_1, \dots, H_k の情報が登録手段 5 に提供され、登録手段 5 により、各表に変更後のライン L' が登録される。ステップ 41 において、ソート手段 6 により、各表に設定されたソート規則 S に基づいて各表がソートされ、変更処理を終了する。

30

40

【0045】

本発明の代表的な実施例においては、1つのラインは3つのワードによって構成されている。そこにおいて、ワードは、文字列、文字列、文字列、整数、整数整数順序対のいずれ

50

かであるか、または浮動小数点数である。1つのラインを登録するときには、このラインは、ラインを構成する3つのワードとソートキーの違いによって同時に4つの表に自動的に登録される。この4つの表は、登録する1つのラインの3つのワードが左から順に a , b , c である場合に、(b , H)、(b , M)、(a , h)、(c , m)、と表される。ここで、(b , H) は、中央のワードが b であるラインの集まりで、左のワードがソートキーである表を表す。(b , M) は、中央のワードが b であるラインの集まりで、右のワードがソートキーである表を表す。(a , h) は、左のワードが a であるラインの集まりで、中央のワードがソートキーである表を表す。(c , m) は、右のワードが c であるラインの集まりで、中央のワードがソートキーである表を表す。後述するように、例えば、1つのライン「 customer 1 TRACE TRECE 800 - 1 」をデータベースに登録する場合には、このラインは、4つの表(TRACE TRACE , H)、(TRACE TRACE , M)、(customer 1 , h)、(800 - 1 , m) にそれぞれ登録される。また、1つのラインを削除する場合には、そのラインが登録されている4つの表すべてから自動的にそのラインが削除される。さらに、1つのラインの内容を変更する場合には、ラインを一旦4つの表から削除し、そのラインを変更し、変更した新たなラインを登録し直す。つまり、1つのラインを新たに登録すると最大4つの表が新たに作られる。

【 0 0 4 6 】

[具体的な実施例の内容]

以下には、本発明による具体的な実施例として、図1に示す構成を有するデータベース装置によって離散事象シミュレーションを行う例を説明する。なお、図1のデータベース装置を構成する手段(機能)とその作用は、前述した通りであるため、以下には、本実施例の特徴のみを説明する。

【 0 0 4 7 】

実施例を説明する前提として、離散事象シミュレーションについて説明する。離散事象シミュレーションにおいては、シミュレートするモデルの初期状態をデータベースに記述する。そして、データベースにある将来生起する事象リストのうち最も早く生起する事象の時刻に時間を進め、その事象を行う。ここで、事象を行うとは、その事象が起きることによってもたらされるモデルの変化を、モデルを記述しているデータベースに反映させることである。つまり、図6に示すように、シミュレーションの実行は、事象によってデータベースを書き換えることによって行われる。

【 0 0 4 8 】

本実施例では、以上のような離散事象シミュレーションを、図7に示すようなシミュレーションモデルに対して実行する。図7においては、窓口が1つしかない銀行のモデルを想定しており、客が窓口に来た順にサービスを受けてサービスが終了すると去っていく場合の、ある時点の状態を示している。この場合、複数の客は、異なる時間に順次到着し、前の客がサービス中ならば、前の客のサービスが終わるまで列に並んでおり、その客自身に対する窓口のサービスが終了した時点で退去するものとする。このモデルでは、客が5人いる。

【 0 0 4 9 】

図8は、図7のモデルの状態を表す全てのラインを示す表であり、これらのラインを、後述するような条件に従って登録すると、図9～12に示すような多数の表からなるデータベースが作成される。言い換えれば、図9～12は、図7のモデルの状態を表すデータベース全体を示している。このデータベースを構成する1つの表を図13に例示する。図13に示すように、表は、ある条件で集めた1つまたはそれ以上のライン集合からなる。図13のライン集合を構成する1つのラインを図14に例示する。図14に示すように、1つのラインは3つのワードからなる。

【 0 0 5 0 】

本実施例においては、図15の表に示すような4つの型のワードが、使用可能なワードとして設定されている。図16は、図15の4つの型のワードの例をそれぞれ示している。

10

20

30

40

50

すなわち、型 1 のワードは 2 つの文字列の対で、型 2 は文字列、整数の対で、型 3 は 2 つの整数の対である。また、型 4 のワードは対ではなく、浮動小数点数 1 個でワードをなす。

【 0 0 5 1 】

ワード間の整列の順序は次のように決定されている。すなわち、基本的に、小さい方から大きい方に向かって上から下に並べるものとする。そして、型 1、2、3 は、型 4 より小さいものとする。また、型 1、2、3 の間の比較は、まず、ワードの対の左の要素を比較して決める。左の要素が同じ場合は、右の要素の比較によって決める。整数、浮動小数点数の間の比較は、それらの数の大小関係をそのまま使う。さらに、文字列と整数の間の比較は、文字列の方が小さいものとし、そして、これらの比較結果に従って、小さい方が上

10

【 0 0 5 2 】

本実施例において、ラインは、単なる 3 つのワードの集まりではなく、その順序が極めて重要である。本実施例では、図示された表において、図中左のワードから右に向かって、それぞれ、左のワード、中央のワード、右のワードと称する。本実施例においては、1 つのラインがデータベースに登録される時、ラインの内容、整列の基準となるワードを、左、中央、右のどのワードにしているかによって、自動的に同時に 4 つの表に登録される。登録すべき表が存在しなければ、表を新たに作成し登録する。

【 0 0 5 3 】

このような 1 つのラインに登録する 4 つの表を表すとき、 x を任意のワードとして、 (x, H) 、 (x, M) 、 (x, h) 、 (x, m) というような記法を使用する。この場合、 (x, H) は、中央のワードが x であるライン集合からなり、左のワードによってラインを整列した表である。 (x, M) は、中央のワードが x であるライン集合からなり、右のワードによってラインを整列した表である。 (x, h) は、左のワードが x であるライン集合からなり、中央のワードによってラインを整列した表である。 (x, m) は、右のワードが x であるライン集合からなり、中央のワードによってラインを整列した表である。たとえば、図 14 に示す 1 つのライン「customer 1 TRACE TRACE E 800 - 1」に登録する 4 つの表は、 $(TRACE TRACE, H)$ 、 $(TRACE TRACE, M)$ 、 $(customer 1, h)$ 、 $(800 - 1, m$

20

30

【 0 0 5 4 】

本実施例のデータベースは、このように、1 つのラインに対して 4 つの表が用意されるので、図 9 ~ 12 に示すように、多数の表が集まって構成される。また、本実施例において、モデルの状態の記述は、主に以下の 4 種類のラインからなる。

【 0 0 5 5 】

第 1 に、事象の状態を表すラインがある。将来生起する事象リストである。これを表すラインの右のワードが、事象の種類を表す。本実施例では、事象は 2 つあり、到着事象と退去事象である。以下には、これらの事象をそれぞれ、「AR」と「LV」で表す。すなわち、到着事象「AR」は、客が銀行に到着することを実行し、退去事象「LV」は客が銀行から去ることを実行する。また、中央のワードは、既に実行が終了した事象が将来起こる事象かを表すワードであり、「PAST EVENT」または「FUTURE EVENT」である。左のワードは、事象生起時刻を表している。たとえば、ライン「sec 1000 FUTURE EVENT AR 0」は、これから起こる事象であり、生起時刻は 1000 sec で、起こる事象は到着事象「AR」であることを示す。

40

【 0 0 5 6 】

第 2 に、このモデルにいる 5 人の客の状態と客のサービス時間を表すラインがある。このラインの左のワードは、客の ID 番号であり、中央のワードは客の状態を表し、右のワードでサービス時間を表す。サービス時間とは、客が窓口に入ったときサービスを終えるのに必要な時間のことである。客の状態は 4 種類あり、それを図 21 に表す。この図 21 に

50

示すように、客がまだ銀行に来ていない状態（未到着状態）を「COMING IN」と表す。客が銀行に到着しているが、窓口でサービスを受けられず、列に並んでいる状態を「IN LINE」と表す。客がまさに窓口でサービスを受けている状態を「BEING SERVED」と表す。客がサービスを終了し銀行を去った状態を「GONE AWAY」と表す。たとえば、ライン「CUSTOMER 1 BEING SERVED sec 400」は、ID番号1の客（客1と呼ぶ。）の状態が、「BEING SERVED」（サービスを受けている）状態であり、客1が必要なサービス時間は、400 secであることを示す。

【0057】

第3に、客の追跡情報を格納するラインがある。このラインには、左に客のID番号を表すワードがあり、中央にこのラインが客の追跡情報であることを表すワード「TRACE TRACE」がある。右のワードの左の要素は、その客が銀行に到着した時刻を表す。右のワードの右の要素は、その客がサービスを終了し銀行を去った時刻を表す。これらの要素には、初期状態では負の数「-1」を入力する。

【0058】

第4に、銀行でサービスを終えて去っていった客の、銀行に滞在していた平均時間を示すラインがある。このラインは、中央に、「MEAN RESPONSE」というワードを持っている。左のワードの左の要素は、去った客の合計滞在時間を表し、左のワードの右の要素は、去った客の数を表す。また、右のワードは平均滞在時間（浮動小数点数）を表す。

【0059】

本実施例においては、ラインの3つのワードのいずれかを変更すると、自動的に登録すべき4つの表に登録され整列される。たとえば、図14に示すライン「customer 1 TRACE TRACE 800 -1」の右のワードを「800 1200」に変更した場合を想定する。この場合、変更前のライン「customer 1 TRACE TRACE 800 -1」は、前記の4つの表（TRACE TRACE, H）、（TRACE TRACE, M）、（customer 1, h）、（800 -1, m）からすべて消去される。そして、新しいライン「customer 1 TRACE TRACE 800 1200」が、新たに4つの表（TRACE TRACE, H）、（TRACE TRACE, M）、（customer 1, h）、（800 1200, m）に登録される。この場合、登録される4つの表のうち、先の3つの表は変更前と変わらないが、表の中でのラインの位置が変わり得る。

【0060】

また、本実施例において、表の検索をする場合には、（x, A）の表と指定することによって、表を検索することができる。ここで、xはワード、AはH, M, h, mのいずれかである。図13は、（customer 1, h）の表である。また、表（customer 1, h）の何番目のラインと指定して、ラインを検索することもできる。さらに、ラインの内容によってラインを検索することもできる。そしてまた、表（customer 1, h）のうち、右のワードが「sec 350」のラインを検索することができる。加えて、検索したラインが、その表の中でどの位置にあるを決定することができる。

【0061】

以下には、本実施例における具体的なシミュレーション操作について説明する。

【0062】

まず、図9～12に示すデータベースを用いて、次に生起する事象を検索する。この場合、次に起きる事象は、図9に示す表のうち、将来生起する事象を時刻順に表す表（FUTURE EVENT, H）の一番上のライン「sec 1000 FUTURE EVENT AR 0」に記述されている。このラインは、時刻1000 secに到着事象「AR」が起ることを表している。そして、この時刻1000 secにおける到着事象「AR」が生起した場合には、この到着事象「AR」により、図9～12に示されるデータが次のようにして書き換えられる。

10

20

30

40

50

【0063】

到着事象「AR」は、未到着である客のうち、ID番号の最も小さい客を探して、銀行に到着させる。未到着である客についてのラインは、表（COMING IN, H）に登録されており、この表は、客のID番号を表す左のワードをソートキーとして整列されている。この未到着の客をID番号順に表す表（COMING IN, H）を図22に示す。この表（COMING IN, H）の一番上のラインを検索すると、その左のワードから、客4が得られる。したがって、客4を到着させ、列に並ばせる。すなわち、客4の状態を、「COMING IN」から「IN LINE」に書き換え、ライン「customer 4 COMING IN sec 300」を、ライン「customer 4 IN LINE sec 300」に変更する。この結果、到着事象「AR」実行前のライン「customer 4 COMING IN sec 300」は、到着事象「AR」実行前にそれが登録されていた4つの表から削除され、新たに、到着事象「AR」実行後のライン「customer 4 IN LINE sec 300」が、対応する4つの表に登録される。図23の表（IN LINE, H）は、到着事象「AR」実行後における待ち行列中の客をID番号順に表している。客4が加わって客2、3、4の順に並んでいることがわかる。

10

【0064】

次に、客4の追跡情報に関するラインを検索し、銀行到着時刻を記録する。この場合、追跡情報は、表（TRACE TRACE, H）に示される。この表の中で、左のワードが「customer 4」であるラインを検索すると、ライン「customer 4 TRACE TRACE - 1 - 1」が得られる。このライン「customer 4 TRACE TRACE - 1 - 1」の右のワードの左の要素を、初期値「- 1」からこの到着事象「AR」の時刻値「1000」に書き換え、ライン「customer 4 TRACE TRACE 1000 - 1」に変更する。この結果、到着事象「AR」実行前のライン「customer 4 TRACE TRACE - 1 - 1」は、到着事象「AR」実行前にそれが登録されていた4つの表から削除され、新たに、到着事象「AR」実行後のライン「customer 4 TRACE TRACE 1000 - 1」が、客の追跡情報を表す表（TRACE TRACE, H）、（TRACE TRACE, M）を含む4つの表に登録される。

20

30

【0065】

続いて、この時刻1000 secにおける到着事象「AR」を表している前記のライン「sec 1000 FUTURE EVENT AR 0」の中央のワードを、「FUTURE EVENT」から「PAST EVENT」に変更する。この結果、到着事象「AR」実行前のライン「sec 1000 FUTURE EVENT AR 0」は、将来生起する事象を客のID番号順に表す表（FUTURE EVENT, H）と将来生起する事象を事象の種類順に表す表（FUTURE EVENT, M）を含む4つの表から削除され、新たに、到着事象「AR」実行後のライン「sec 1000 PAST EVENT AR 0」が、過去に生起した事象を表す表（PAST EVENT, H）、（PAST EVENT, M）を含む4つの表に登録される。

40

【0066】

図24は、以上のような時刻1000 secにおける到着事象「AR」実行後のモデルの状態を表している。図25は、図24のモデルの状態を表す全てのラインを示す表であり、各ラインをそれぞれ対応する4つの表に登録することにより、図26～29に示すような多数の表からなるデータベースが作成される。この図26～29に示すデータベースにおいて、次に生起する事象は、図26に示す表のうち、将来生起する事象を時刻順に表す表（FUTURE EVENT, H）の一番上のライン「sec 1200 FUTURE EVENT LV 0」に記述されている。このラインは、時刻1200 secに退去事象「LV」が起ることを表している。そして、この時刻1200 sec

50

における退去事象「LV」が生じた場合には、この退去事象「LV」により、図26～29に示されるデータが次のようにして書き換えられる。

【0067】

退去事象「LV」は、まず、現在窓口にいる客を、表(BEING SERVED, M)から検索する。この窓口にいる客をサービス時間順に表す表(BEING SERVED, M)には、図27に示されるように、1つのライン「customer 1 BEING SERVED sec 400」のみが登録されている。このラインから、現在窓口にいる客が、客1であることがわかる。したがって、客1を銀行から退去させる。すなわち、このライン「customer 1 BEING SERVED sec 400」において、客の状態を示す中央のワードを、「BEING SERVED」から「GONE AWAY」に書き換え、ライン「customer 1 BEING SERVED sec 400」を、ライン「customer 1 GONE AWAY sec 400」に変更する。この結果、退去事象「LV」実行前のライン「customer 1 BEING SERVED sec 400」は、退去事象「LV」実行前にそれが登録されていた4つの表から削除され、新たに、退去事象「LV」実行後のライン「customer 1 GONE AWAY sec 400」が、対応する4つの表に登録される。

【0068】

次に、列の先頭で待っている客を窓口に入れる。待ち行列中の客をID番号順に表す表(IN LINE, H)の一番上のラインを検索する。この表(IN LINE, H)の一番上のラインは、図26に示されるように、ライン「customer 2 IN LINE sec 300」である。このラインから、次に窓口に入る客が、客2であることがわかる。したがって、客2を窓口に入れる。すなわち、このライン「customer 2 IN LINE sec 300」において、客の状態を示す中央のワードを「IN LINE」から「BEING SERVED」に書き換え、ライン「customer 2 IN LINE sec 300」を、ライン「customer 2 BEING SERVED sec 300」に変更する。この結果、退去事象「LV」実行前のライン「customer 2 IN LINE sec 300」は、退去事象「LV」実行前にそれが登録されていた4つの表から削除され、新たに、退去事象「LV」実行後のライン「customer 2 BEING SERVED sec 300」が、対応する4つの表に登録される。

【0069】

以上のような操作の結果、現在窓口にいる客を表す表(BEING SERVED, H)、(BEING SERVED, M)と、待ち行列中の客を表す表(IN LINE, H)、(IN LINE, M)の内容が変更される。また、退去した客に関するラインを、前述のように変えた結果、新たに、退去した客を示す表(GONE AWAY, H)、(GONE AWAY, M)が作成される。図30は、このような時刻1200 secにおける退去事象「LV」実行前の客の状態を表す3つの表(BEING SERVED, H)、(IN LINE, H)、(COMING IN, H)を示しており、図31は、時刻1200 secにおける退去事象「LV」実行後の客の状態を表す4つの表(GONE AWAY, H)、(BEING SERVED, H)、(IN LINE, H)、(COMING IN, H)を示している。

【0070】

本実施例において、上述のように、新たに窓口に客2が入ると、その客2が窓口を去る退去時刻は、「現時刻+サービス時間」によって決定される。すなわち、次の退去事象「LV」が生起する時刻である。この例において、客2の退去時刻は、「1200+300」から、1500 secになる。次の退去事象「LV」が1500 secに生起するので、この事象に関するラインを新たに登録する。すなわち、ライン「sec 1500 FUTURE EVENT LV 0」を、将来生起する事象の表(FUTURE EVENT, H)、(FUTURE EVENT, M)を含む4つの表に登録する。

また、時刻 1200 sec における退去事象「LV」を表している前記のライン「sec 1200 FUTURE EVENT LV 0」の中央のワードを、「FUTURE EVENT」から「PAST EVENT」に変更する。この結果、退去事象「LV」実行前のライン「sec 1200 FUTURE EVENT LV 0」は、将来生起する事象の表（FUTURE EVENT, H）、（FUTURE EVENT, M）を含む4つの表から削除され、新たに、退去事象「LV」実行後のライン「sec 1200 PAST EVENT LV 0」が、過去に生起した事象の表（PAST EVENT, H）、（PAST EVENT, M）を含む4つの表に登録される。そして、以上のような操作の結果、将来生起する事象を時刻順に表す表（FUTURE EVENT, H）の内容は、図32のようになる。

10

【0071】

さらに、客1の追跡情報に関するラインを検索し、銀行退去時刻を記録する。すなわち、客の追跡情報を客のID番号順に表す表（TRACE TRACE, H）の中で、客1に関するライン「customer 1 TRACE TRACE 800 - 1」の右のワードの右の要素を、初期値「- 1」からこの退去事象「LV」の時刻値「1200」に書き換え、ライン「customer 1 TRACE TRACE 800 - 1」を、ライン「customer 1 TRACE TRACE 800 1200」に変更する。この結果、退去事象「LV」実行前のライン「customer 1 TRACE TRACE 800 - 1」は、退去事象「LV」実行前にそれが登録されていた4つの表から削除され、新たに、退去事象「LV」実行後のライン「customer 1 TRACE TRACE 800 1200」が、客の追跡情報の表（TRACE TRACE, H）、（TRACE TRACE, M）を含む4つの表に登録される。そして、以上のような操作の結果、客の追跡情報を客のID番号順に表す表（TRACE TRACE, H）の内容は、図33のようになる。

20

【0072】

図34は、以上のような時刻1200 sec における退去事象「LV」実行後のモデルの状態を表している。図35は、図34のモデルの状態を表す全てのラインを示す表であり、各ラインをそれぞれ対応する4つの表に登録することにより、図36～39に示すような多数の表からなるデータベースが作成される。

【0073】

なお、本発明は、前記実施例に限定されるものではなく、例えば、1つのラインを構成するワードの数は3つに限らず、4つ以上のワードによって1つのラインを構成することも可能であり、逆に、2つのみのワードによって1つのラインを構成することも可能である。また、表のソート規則およびそれに関連する一定の登録条件の設定内容についても自由に設定可能である。例えば、ある位置のワードでソートするだけの単純なソート規則、あるいは逆に、複数のワードをソートキーにするソート規則などを使用することが可能であり、このようなソート規則に応じて、登録条件の内容を適宜設定することができる。

30

【0074】

より具体的に、前記実施例の変形例として、3つのワードで構成される1つのラインを登録する表として、前記実施例における4つの表に加え、さらに、左のワードが同じであるライン集合からなり、右のワードをソートキーとする第5の表を対応させるように構成する実施例などが考えられる。また、1つのラインを5つのワードで構成し、1つのラインを登録する表として、第1のワードが同じであるライン集合からなり、第2のワードをソートキーとする第1の表と、第2のワードが同じであるライン集合からなり、第3のワードを第1ソートキー、第4のワードを第2ソートキーとする第2の表、およびその他の数種類の表を対応させるように構成する実施例なども考えられる。

40

【0075】

ところで、本発明において、ラインをソートするソート手段としては、各種の手段を使用可能であるが、例えば、ソート、検索、登録、および削除可能な二分探索木を使用することが可能である。以下には、この二分探索木について簡単に解説する。

50

【0076】

まず、二分木といくつかの言葉を定義する。図40のように、二分木はノード（印）とノードの間の親子関係を示す枝（ノ、\）からなり、どのノードも、子ノードを最大2つしか持たず、かつ、ある1つのノード以外のどのノードも、親ノードをただ1つだけ持つ。左下の子ノードを左子ノード、右下の子ノードを右子ノードと呼ぶ。親ノードを持たない唯一のノードを根ノードと呼ぶ。図40において、ノード（1）はノード（2）の親ノードであり、ノード（2）はノード（1）の右子ノードである。ノード（3）は二分木の根ノードである。ノードの左（右）の子ノードの下の構造全体を左（右）部分木と言う。図40において、部分木（4）はノード（1）の右部分木である。

【0077】

10

本発明において利用する二分木は、ソート規則の定められた対象があり、二分木のノードがその対象（前記実施例においては、データベースのライン）を指示していて、二分木の任意のノードが指示している対象が、そのノードの左（右）部分木のどのノードが指す対象と比して大きい（小さい）ような二分木である。等しい場合を左に含めるか右に含めるかは二分探索木を使う事情による。ただし、小さい方が表の上であり、大きい方が表の下にある。

【0078】

たとえば、図41は、整数をソートしている二分探索木である。なお、この図41では、図面の簡略化の目的で、ノードが指示している対象である整数を直接ノードに書き込んでいる。この図41において、ノード（1）が指す整数「54」は、その右部分木（2）の

20

どのノードの指示する整数「60」、「58」、「71」、「56」よりも小さい。

【0079】

図42は、このような二分探索木での検索方法の一例を示すフローチャートである。このフローチャートによって、たとえば、図41の二分探索木で整数「9」を検索する場合には、次のような手順の処理を行うことになる。この処理について、図41を参照して説明する。なお、以下の説明において、Nは現在のノードを意味する。まず、図41の根ノード（3）をNとする。この場合、検索する整数「9」はNが指示する整数「46」より小さい。したがって、Nの左子ノード（4）を新たにNにする。この場合、検索する整数「9」はNが指示する整数「13」より小さい。したがって、Nの左子ノード（5）を新たにNとする。この場合、Nが指示する整数は「9」であり、このノードが、検索目的のノ

30

【0080】

図43は、二分探索木での登録方法の一例を示すフローチャートである。このフローチャートによって、たとえば、図41の二分探索木に対して、整数「48」を指示するノードを登録する場合には、次のような手順の処理を行うことになる。この処理について、図44を参照して説明する。まず、図44の根ノード（1）をNにする。この場合、登録する整数「48」は、Nが指示する整数「46」より大きい。よって、Nの右子ノード（2）を新たにNとする。この場合、登録する整数「48」はNが指示する整数「54」より小さい。したがって、Nの左子ノード（3）を新たにNにする。この場合、登録する整数「48」はNが指示する整数「49」より小さいが、現在のノードNの左には子ノードがないので、Nを親ノードとして、整数「48」を指示する左子ノード（4）を作成する。

40

【0081】

図45は、二分探索木での削除方法の一例を示すフローチャートである。このフローチャートによって、次のような処理を行うことができる。すなわち、まず、ノードAを削除する場合、そのノードAが子ノードを全く持たない場合には、図46に示すように、そのままノードAを削除する。また、ノードAが左（右）子ノードだけしか持たない場合には、図47に示すように、このノードAを消して、左（右）部分木をノードAの位置に付け替える。一方、ノードAが左右の両子ノードを持つ場合には、図48に示すように、ソート順においてノードAの1つ前のノード（これはノードAの左部分木の中で最も右にあるノードである。）をNとして（図48の左の状態）、ノードAとノードNを入れ替える（図

50

48の操作 1 : 左 中央の状態)。このように入れ替えた際に、ノード A が左部分木を持たない場合には、単にノード A を消す。ノード A が左部分木を持つならば、その左部分木をノード A の位置に付け替える (図 48 の操作 2 : 中央 右の状態)。

【0082】

なお、ラインをソートするソート手段として、このような二分探索木を用いることは有効であるが、前述の通り、ラインのソート、検索、登録、および削除である限り、他のソート手段を使用することも同様に可能である。

【0083】

続いて、本発明におけるデータベースへの機能の実装例を、図 49 によって説明する。この例では、図 49 の中央部分に 6 つのライン xuw 、 vxz 、 wxy 、 xxx 、 yxx 、 ywx が示されている。ここで、 u 、 v 、 w 、 x 、 y 、 z は、それぞれ異なるワードを表す。ラインの検索に際しては、各ワードに対応する 4 つの異なる表を使用できる。たとえば、ワードを x とすると、4 つの表は、 (x, H) 、 (x, M) 、 (x, h) 、 (x, m) で表される。これらの表は、図 49 中において、二分探索木：木 (1) ~ (4) として表現されている。この場合、木の各ノードは、ラインとそのラインのソートキーになっているワードを指示している。たとえば、木 (1) の根のノードはライン wxy を指示し、さらにソートキーであるワード w を指示している。逆にラインも対応するノードを指示している。ある木のノードが指示しているラインは、その木が表現する表に含まれることを意味する。たとえば表 (x, H) に対応する木 (1) には、ライン xxx を指示するノードがあるので、ライン xxx は、表 (x, H) に含まれている。

【0084】

表 (x, H) は左のワードがソートキーなので、この表を表現する二分探索木の各ノードは、対応するラインと左のワードを指示している。すなわち、各表の各ワードは、対応するラインを指示するとともに、 (x, M) なら右、 (x, h) 、 (x, m) なら中央のワードも指示している。たとえば、図 49 のライン xxx は、4 つの表を表現する 4 つの二分探索木のノードによって指示されている。また逆に、ライン xxx は、このライン xxx を指示するノードを指示している。つまり、このライン xxx を指定することにより、このライン xxx を含む 4 つの表を表現している 4 つの二分探索木を見つけられる。

【0085】

以上のような構造を有する図 49 のデータベースにおいて、ラインを削除する方法を説明する。データベース上からライン xxx を削除する場合には、このライン xxx を含む 4 つの表 (x, H) 、 (x, M) 、 (x, h) 、 (x, m) からライン xxx を削除する。この場合、図 49 上の木 (1) ~ (4) のライン xxx を指示しているノードを削除しなければならない。ノードの削除は、二分探索木に関して前述した手順に従う。その結果、図 50 に示すような、ライン xxx のないデータベースが得られる。

【0086】

逆に、図 50 に示すようなデータベースに対して、ライン xxx を登録するときは、4 つの表 (x, H) 、 (x, M) 、 (x, h) 、 (x, m) にそれぞれライン xxx を登録しなければならない。この場合、これらの 4 つの表に対応する木 (1) ~ (4) にライン xxx を指示するノードを登録する。ノードの登録の手順は、二分探索木に関して前述した手順に従う。

【0087】

さらに、図 49 のデータベースにおける任意のラインの検索は、図 51 のようにして行うことができる。この図 51 は、一例として、左と中央にワード x を持つライン $xx?$ を 1 つ検索し、右のワードを検索する場合の手順を示している。このライン $xx?$ を含む可能性のある表は、 (x, H) 、 (x, M) 、 (x, h) 、 (x, m) である。中央のワードが x で、左のワードをソートキーとする表 (x, H) を使用することにより、目的のラインを探索できる。この図 51 上で、まず、ワード x を起点とする矢印 (1) をたどると、矢印 (1) は表 (x, H) を指示している。この表のラインは、整列順を保持するため、二分探索木のノードに格納されている。表 (x, H) から矢印 (2) をたどると、その二

10

20

30

40

50

分探索木：木（３）の根を指示している。木の根を指示することは、木自体を指示することに等しい。また逆に、表（ x, H ）のラインは、木（３）の各ノードから指示されている。

【００８８】

続いて、木（３）の根から矢印（４）をたどると、根のノードが指示しているラインは $w \times y$ である。この場合、ライン $w \times y$ と検索するライン $x \times ?$ の大きさを比較すると $x \times ?$ の方が大きい（つまり表の下、木のノードの右にある）。そこで、木（３）の根の右の枝（５）を下がり、その先のノードが矢印（６）によって指示するライン $y \times x$ と比較する。この場合、 $x \times ?$ の方が小さいので、左の枝（７）を下がる。その枝（７）の先のノードが矢印（８）によって指示するラインは $x \times x$ であり、検索しているラインに該当する。したがって、左と中央にワード x を持つライン $x \times ?$ の右のワードは x であることがわかる。

10

【００８９】

【発明の効果】

以上説明したように、本発明においては、ラインと表との対応関係を示す一定の登録条件を予め設定しておくことにより、この登録条件に基づいて、１つのラインを登録する複数の表を自動的に指定することができるため、新たに登録するラインまたは内容を変更するラインとすでに登録されているラインとの関連を容易に把握可能であり、表とラインの管理が簡略で、設計時に想定されなかった使い方に対しても柔軟に対処でき、操作性および能率に優れたデータベース装置を提供することができる。

20

【図面の簡単な説明】

【図１】本発明によるデータベース装置の基本的な機能構成の一例を示す機能ブロック図。

【図２】本発明によるデータベース装置の一般的な登録処理手順の一例を示すフローチャート。

【図３】本発明によるデータベース装置の一般的な検索処理手順の一例を示すフローチャート。

【図４】本発明によるデータベース装置の一般的な削除処理手順の一例を示すフローチャート。

【図５】本発明によるデータベース装置の一般的な変更処理手順の一例を示すフローチャート。

30

【図６】離散事象シミュレーションの手順を示す説明図。

【図７】本発明のデータベース装置によって離散事象シミュレーションを行う一実施例において、シミュレートするモデルの１つの状態を示す模式図。

【図８】図７のモデルの状態を表す全てのラインを示す表。

【図９】図８のラインから作成されるデータベースの一部を構成する複数の表を示す図。

【図１０】図８のラインから作成されるデータベースの一部を構成する複数の表を示す図。

【図１１】図８のラインから作成されるデータベースの一部を構成する複数の表を示す図。

40

【図１２】図８のラインから作成されるデータベースの一部を構成する複数の表を示す図。

【図１３】図９～１２のデータベースを構成する１つの表を示す図。

【図１４】図１３のライン集合を構成する１つのラインを示す図。

【図１５】本発明において使用可能なワードの４つの型を示す表。

【図１６】図１５のワードの例を示す表。

【図１７】図１４のラインを登録する１つの表を示す図。

【図１８】図１４のラインを登録する１つの表を示す図。

【図１９】図１４のラインを登録する１つの表を示す図。

【図２０】図１４のラインを登録する１つの表を示す図。

50

【図 2 1】図 7 のモデルにいる客の 4 種類の状態を示す表。

【図 2 2】図 9 ~ 1 2 のデータベースを構成する表のうち、未到着の客を I D 番号順に表す表。

【図 2 3】図 9 ~ 1 2 のデータベースを用いた離散事象シミュレーションによる到着事象実行後の待ち行列中の客を I D 番号順に表す表。

【図 2 4】図 7 のモデルの、離散事象シミュレーションによる到着事象実行後における状態を示す模式図。

【図 2 5】図 2 4 のモデルの状態を表す全てのラインを示す表。

【図 2 6】図 2 5 のラインから作成されるデータベースの一部を構成する複数の表を示す図。

10

【図 2 7】図 2 5 のラインから作成されるデータベースの一部を構成する複数の表を示す図。

【図 2 8】図 2 5 のラインから作成されるデータベースの一部を構成する複数の表を示す図。

【図 2 9】図 2 5 のラインから作成されるデータベースの一部を構成する複数の表を示す図。

【図 3 0】図 2 6 ~ 2 9 のデータベースを用いた離散事象シミュレーションによる退去事象実行前の客の状態を表す 3 つの表を示す図。

【図 3 1】図 2 6 ~ 2 9 のデータベースを用いた離散事象シミュレーションによる退去事象実行後の客の状態を表す 4 つの表を示す図。

20

【図 3 2】図 2 6 ~ 2 9 のデータベースを用いた離散事象シミュレーションによる退去事象実行後の時点での、将来生起する事象を時刻順に表す表の内容を示す図。

【図 3 3】図 2 6 ~ 2 9 のデータベースを用いた離散事象シミュレーションによる退去事象実行後の時点での、客の追跡情報を客の I D 番号順に表す表の内容を示す図。

【図 3 4】図 2 4 のモデルの、離散事象シミュレーションによる退去事象実行後における状態を示す図。

【図 3 5】図 3 4 のモデルの状態を表す全てのラインを示す表。

【図 3 6】図 3 5 のラインから作成されるデータベースの一部を構成する複数の表を示す図。

【図 3 7】図 3 5 のラインから作成されるデータベースの一部を構成する複数の表を示す図。

30

【図 3 8】図 3 5 のラインから作成されるデータベースの一部を構成する複数の表を示す図。

【図 3 9】図 3 5 のラインから作成されるデータベースの一部を構成する複数の表を示す図。

【図 4 0】本発明によるデータベース装置においてソート手段として使用可能な二分探索木の一例を示す図。

【図 4 1】整数をソートしている二分探索木の一例を示す図。

【図 4 2】二分探索木での検索方法の一例を示すフローチャート。

【図 4 3】二分探索木での登録方法の一例を示すフローチャート。

40

【図 4 4】図 4 1 の二分探索木に対して、新しいノードを登録する手順とそれを登録した二分探索木を示す図。

【図 4 5】二分探索木での削除方法の一例を示すフローチャート。

【図 4 6】図 4 5 の方法による削除操作の一例を示す説明図。

【図 4 7】図 4 5 の方法による削除操作の一例を示す説明図。

【図 4 8】図 4 5 の方法による削除操作の一例を示す説明図。

【図 4 9】本発明に従って機能を実装したデータベースの一例を示す説明図。

【図 5 0】図 4 9 のデータベースから 1 つのラインを削除して得られたデータベースを示す説明図。

【図 5 1】図 4 9 のデータベースにおける任意のラインの検索手順を示す説明図。

50

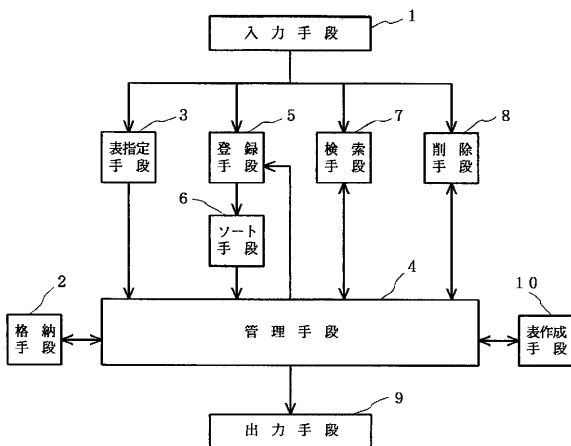
【図52】データベースに登録されるリレーションの一例を示す表。

【図53】データベースに登録されるリレーションの一例を示す表。

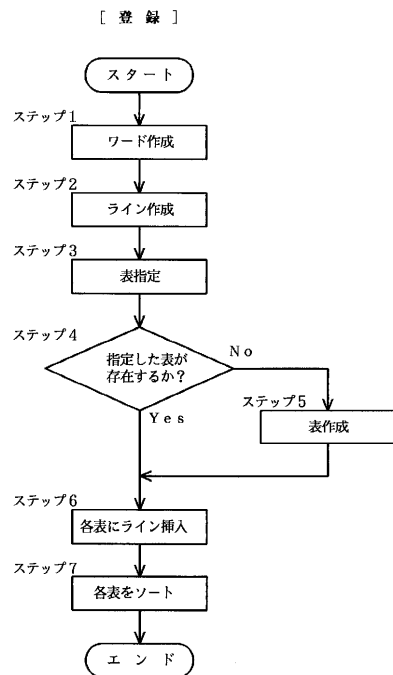
【符号の説明】

- 1 ... 入力手段
- 2 ... 格納手段
- 3 ... 表指定手段
- 4 ... 管理手段
- 5 ... 登録手段
- 6 ... ソート手段
- 7 ... 検索手段
- 8 ... 削除手段
- 9 ... 出力手段
- 10 ... 表作成手段

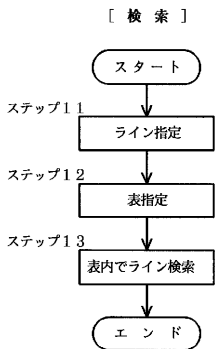
【図1】



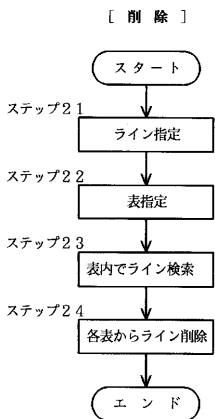
【図2】



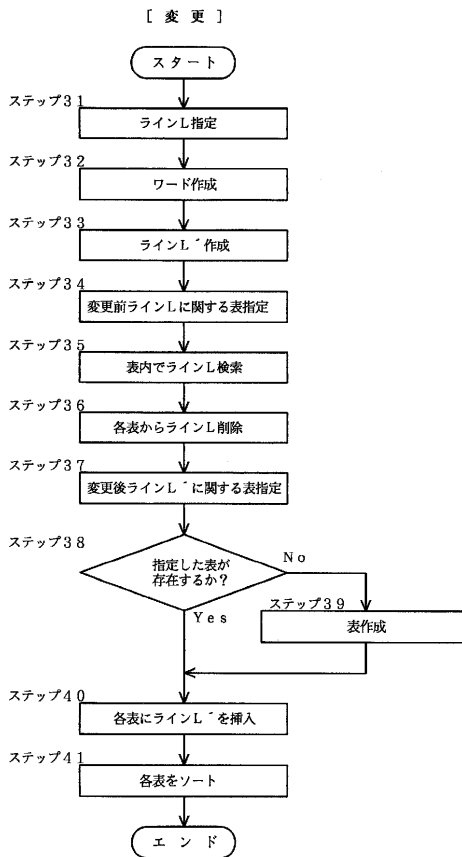
【 図 3 】



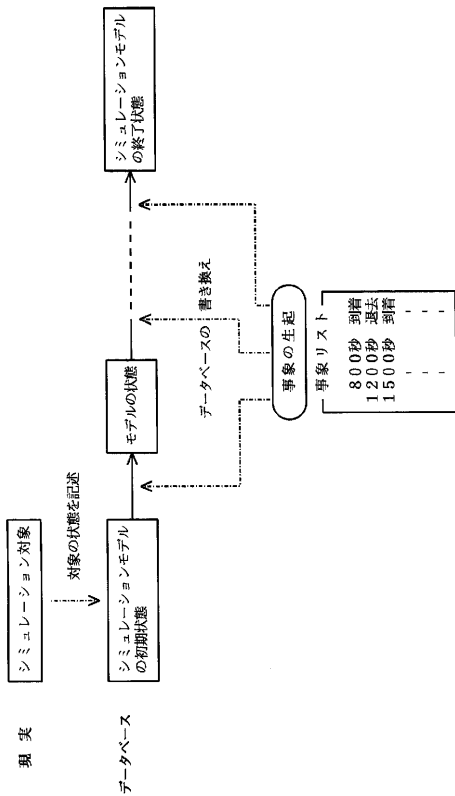
【 図 4 】



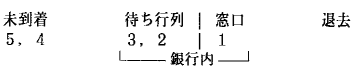
【 図 5 】



【 図 6 】



【 図 7 】



【 図 8 】

sec 1000	FUTURE EVENT	AR	0
sec 1200	FUTURE EVENT	LV	0
sec 2800	FUTURE EVENT	AR	0
sec 3051	FUTURE EVENT	PAUS	0
sec 0	PAST EVENT	DECL	0
sec 800	PAST EVENT	AR	0
sec 840	PAST EVENT	AR	0
sec 930	PAST EVENT	AR	0
sec 931	PAST EVENT	PAUS	0
customer 1	BEING SERVED	SEC	400
customer 2	IN LINE	SEC	300
customer 3	IN LINE	SEC	200
customer 4	COMING IN	SEC	300
customer 5	COMING IN	SEC	250
customer 1	TRACE TRACE	800	-1
customer 2	TRACE TRACE	840	-1
customer 3	TRACE TRACE	930	-1
customer 4	TRACE TRACE	-1	-1
customer 5	TRACE TRACE	-1	-1
0 0	MEAN RESPONSE	0	

【 図 9 】

sec 1000	FUTURE EVENT	AR	0
sec 1200	FUTURE EVENT	LV	0
sec 2800	FUTURE EVENT	AR	0
sec 3051	FUTURE EVENT	PAUS	0
sec 0	PAST EVENT	DECL	0
sec 800	PAST EVENT	AR	0
sec 840	PAST EVENT	AR	0
sec 930	PAST EVENT	AR	0
sec 931	PAST EVENT	PAUS	0
customer 1	BEING SERVED	SEC	400
customer 2	IN LINE	SEC	300
customer 3	IN LINE	SEC	200
customer 4	COMING IN	SEC	300
customer 5	COMING IN	SEC	250
customer 1	TRACE TRACE	800	-1
customer 2	TRACE TRACE	840	-1
customer 3	TRACE TRACE	930	-1
customer 4	TRACE TRACE	-1	-1
customer 5	TRACE TRACE	-1	-1
0 0	MEAN RESPONSE	0	

【 ☒ 1 0 】

sec 1000	FUTURE EVENT	AR 0
sec 2800	FUTURE EVENT	AR 0
sec 1200	FUTURE EVENT	LV 0
sec 3051	FUTURE EVENT	PAUS 0
sec 800	PAST EVENT	AR 0
sec 840	PAST EVENT	AR 0
sec 930	PAST EVENT	AR 0
sec 0	PAST EVENT	DECL 0
sec 931	PAST EVENT	PAUS 0
customer 5	COMING IN	SEC 250
customer 4	COMING IN	SEC 300
customer 3	IN LINE	SEC 200
customer 2	IN LINE	SEC 300
customer 1	BEING SERVED	SEC 400
customer 4	TRACE TRACE	-1 -1
customer 5	TRACE TRACE	-1 -1
customer 1	TRACE TRACE	800 -1
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
0 0	MEAN RESPONSE	0

【 ☒ 1 1 】

customer 1	BEING SERVED	sec 400
customer 1	TRACE TRACE	800 -1
customer 1	WRITE EVENT	x2 2
customer 2	IN LINE	sec 300
customer 2	TRACE TRACE	840 -1
customer 2	WRITE EVENT	x2 2
customer 3	IN LINE	sec 200
customer 3	TRACE TRACE	930 -1
customer 3	WRITE EVENT	x2 2
customer 4	COMING IN	sec 300
customer 4	TRACE TRACE	-1 -1
customer 4	WRITE EVENT	x2 2
customer 5	COMING IN	sec 250
customer 5	TRACE TRACE	-1 -1
customer 5	WRITE EVENT	x2 2
sec 1000	FUTURE EVENT	AR 0
sec 1000	WRITE EVENT	x2 2
sec 1200	FUTURE EVENT	LV 0
sec 1200	WRITE EVENT	x2 2
sec 2800	FUTURE EVENT	AR 0
sec 2800	WRITE EVENT	x2 2
sec 3051	FUTURE EVENT	PAUS 0
sec 3051	WRITE EVENT	x2 2
sec 0	PAST EVENT	DECL 0
sec 0	WRITE EVENT	x2 2
sec 800	PAST EVENT	AR 0
sec 800	WRITE EVENT	x2 2
sec 840	PAST EVENT	AR 0
sec 840	WRITE EVENT	x2 2
sec 930	PAST EVENT	AR 0
sec 930	WRITE EVENT	x2 2
sec 931	PAST EVENT	PAUS 0
sec 931	WRITE EVENT	x2 2
0 0	MEAN RESPONSE	0
0 0	WRITE EVENT	x2 2

【 ☒ 1 2 】

sec 1000	FUTURE EVENT	AR 0
sec 2800	FUTURE EVENT	AR 0
sec 800	PAST EVENT	AR 0
sec 840	PAST EVENT	AR 0
sec 930	PAST EVENT	AR 0
sec 1200	FUTURE EVENT	LV 0
FUTURE EVENT	DEUS 0	PAUS 0
sec 3051	FUTURE EVENT	PAUS 0
sec 931	PAST EVENT	PAUS 0
CONTINUE 0	WRITE EVENT	PAUS 0
x0 0	WRITE EVENT	PAUS 0
x1 1	WRITE EVENT	PAUS 0
x2 2	WRITE EVENT	PAUS 0
x3 3	WRITE EVENT	PAUS 0
sec 0	PAST EVENT	DECL 0
customer 1	TRACE TRACE	800 -1
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
customer 4	TRACE TRACE	-1 -1
customer 5	TRACE TRACE	-1 -1
0 0	MEAN RESPONSE	0

【 ☒ 1 6 】

型1	BEING SERVED
型2	SEC 400
型3	800 -1
型4	540.25

【 ☒ 1 7 】

(TRACE TRACE, H)

customer 1	TRACE TRACE	800 -1
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
customer 4	TRACE TRACE	-1 -1
customer 5	TRACE TRACE	-1 -1

【 ☒ 1 8 】

(TRACE TRACE, M)

customer 4	TRACE TRACE	-1 -1
customer 5	TRACE TRACE	-1 -1
customer 1	TRACE TRACE	800 -1
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1

【 ☒ 1 3 】

customer 1	BEING SERVED	SEC 400
customer 1	TRACE TRACE	800 -1

【 ☒ 1 9 】

(customer 1, h)

customer 1	BEING SERVED	SEC 400
customer 1	TRACE TRACE	800 -1

【 ☒ 1 4 】

customer 1	TRACE TRACE	800 -1
------------	-------------	--------

【 ☒ 2 0 】

(800 -1, m)

customer 1	TRACE TRACE	800 -1
------------	-------------	--------

【 ☒ 1 5 】

型1	文字列	文字列
型2	文字列	整数
型3	整数	整数
型4	浮動小数点数	

【 ☒ 2 1 】

COMING IN	客がまだ銀行にきていない状態
IN LINE	客が銀行で列に並んでいる状態
BEING SERVED	客が窓口でサービスを受けている状態
GONE AWAY	客がサービスを終了し銀行を去った状態

【 2 2 】

customer 4	COMING IN	SEC 300
customer 5	COMING IN	SEC 250

【 2 3 】

customer 2	IN LINE	SEC 300
customer 3	IN LINE	SEC 200
customer 4	IN LINE	SEC 300

【 2 4 】

未到着 待ち行列 | 窓口 退去
 5 4, 3, 2 | 1
 銀行内

【 2 5 】

sec 1200	FUTURE EVENT	LV 0
sec 2800	FUTURE EVENT	AR 0
sec 3051	FUTURE EVENT	PAUS 0
sec 0	PAST EVENT	DECL 0
sec 800	PAST EVENT	AR 0
sec 840	PAST EVENT	AR 0
sec 930	PAST EVENT	AR 0
sec 1000	PAST EVENT	AR 0
sec 1001	PAST EVENT	PAUS 0
customer 1	BEING SERVED	SEC 400
customer 2	IN LINE	SEC 300
customer 3	IN LINE	SEC 200
customer 4	IN LINE	SEC 300
customer 5	COMING IN	SEC 250
customer 1	TRACE TRACE	800 -1
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
customer 4	TRACE TRACE	1000 -1
customer 5	TRACE TRACE	-1 -1
0 0	MEAN RESPONSE	0

【 2 6 】

sec 1200	FUTURE EVENT	LV 0
sec 2800	FUTURE EVENT	AR 0
sec 3015	FUTURE EVENT	PAUS 0

sec 0	PAST EVENT	DECL 0
sec 800	PAST EVENT	AR 0
sec 840	PAST EVENT	AR 0
sec 930	PAST EVENT	AR 0
sec 1000	PAST EVENT	AR 0
sec 1001	PAST EVENT	PAUS 0

customer 1	BEING SERVED	sec 400
------------	--------------	---------

customer 2	IN LINE	sec 300
customer 3	IN LINE	sec 200
customer 4	IN LINE	sec 300

customer 5	COMING IN	sec 250
------------	-----------	---------

customer 1	TRACE TRACE	800 -1
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
customer 4	TRACE TRACE	1000 -1
customer 5	TRACE TRACE	-1 -1

0 0	MEAN RESPONSE	0
-----	---------------	---

【 2 7 】

sec 2800	FUTURE EVENT	AR 0
sec 1200	FUTURE EVENT	LV 0
sec 3051	FUTURE EVENT	PAUS 0

sec 800	PAST EVENT	AR 0
sec 840	PAST EVENT	AR 0
sec 930	PAST EVENT	AR 0
sec 1000	PAST EVENT	AR 0
sec 0	PAST EVENT	DECL 0
sec 1001	PAST EVENT	PAUS 0

customer 5	COMING IN	sec 250
------------	-----------	---------

customer 3	IN LINE	sec 200
customer 2	IN LINE	sec 300
customer 4	IN LINE	sec 300

customer 1	BEING SERVED	sec 400
------------	--------------	---------

customer 5	TRACE TRACE	-1 -1
customer 1	TRACE TRACE	800 -1
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
customer 4	TRACE TRACE	1000 -1

0 0	MEAN RESPONSE	0
-----	---------------	---

【 2 8 】

customer 1	BEING SERVED	sec 400
customer 1	TRACE TRACE	800 -1
customer 1	WRITE EVENT	x2 2

customer 2	IN LINE	sec 300
customer 2	TRACE TRACE	840 -1
customer 2	WRITE EVENT	x2 2

customer 3	IN LINE	sec 200
customer 3	TRACE TRACE	930 -1
customer 3	WRITE EVENT	x2 2

customer 4	IN LINE	sec 300
customer 4	TRACE TRACE	1000 -1
customer 4	WRITE EVENT	x2 2

customer 5	COMING IN	sec 250
customer 5	TRACE TRACE	-1 -1
customer 5	WRITE EVENT	x2 2

sec 1000	PAST EVENT	AR 0
sec 1000	WRITE EVENT	x2 2

sec 1200	FUTURE EVENT	LV 0
sec 1200	WRITE EVENT	x2 2

sec 2800	FUTURE EVENT	AR 0
sec 2800	WRITE EVENT	x2 2

sec 3051	FUTURE EVENT	PAUS 0
sec 3051	WRITE EVENT	x2 2

sec 0	PAST EVENT	DECL 0
sec 0	WRITE EVENT	x2 2

sec 800	PAST EVENT	AR 0
sec 800	WRITE EVENT	x2 2

sec 840	PAST EVENT	AR 0
sec 840	WRITE EVENT	x2 2

sec 930	PAST EVENT	AR 0
sec 930	WRITE EVENT	x2 2

0 0	MEAN RESPONSE	0
0 0	WRITE EVENT	x2 2

【 2 9 】

sec 2800	FUTURE EVENT	AR 0
sec 800	PAST EVENT	AR 0
sec 840	PAST EVENT	AR 0
sec 930	PAST EVENT	AR 0
sec 1000	PAST EVENT	AR 0

sec 1200	FUTURE EVENT	LV 0

FUTURE EVENT	DEUS 0	PAUS 0
sec 3051	FUTURE EVENT	PAUS 0
sec 1001	PAST EVENT	PAUS 0
CONTINUE 0	WRITE EVENT	PAUS 0
x0 0	WRITE EVENT	PAUS 0
x1 1	WRITE EVENT	PAUS 0
x2 2	WRITE EVENT	PAUS 0
x3 3	WRITE EVENT	PAUS 0

sec 0	PAST EVENT	DECL 0

customer 4	TRACE TRACE	1000 -1

customer 1	TRACE TRACE	800 -1

customer 2	TRACE TRACE	840 -1

customer 3	TRACE TRACE	930 -1

customer 5	TRACE TRACE	-1 -1

0 0	MEAN RESPONSE	0

【 3 0 】

customer 1	BEING SERVED	SEC 400

customer 2	IN LINE	SEC 300
customer 3	IN LINE	SEC 200
customer 4	IN LINE	SEC 300

customer 5	COMING IN	SEC 250

【 3 1 】

customer 1	GONE AWAY	SEC 400

customer 2	BEING SERVED	SEC 300

customer 3	IN LINE	SEC 200
customer 4	IN LINE	SEC 300

customer 5	COMING IN	SEC 250

【 3 6 】

sec 1500	FUTURE EVENT	LV 0
sec 2800	FUTURE EVENT	AR 0
sec 3051	FUTURE EVENT	PAUS 0

sec 0	PAST EVENT	DECL 0
sec 800	PAST EVENT	AR 0
sec 840	PAST EVENT	AR 0
sec 930	PAST EVENT	AR 0
sec 1000	PAST EVENT	AR 0
sec 1000	PAST EVENT	AR 0
sec 1200	PAST EVENT	LV 0
sec 1201	PAST EVENT	PAUS 0

customer 1	GONE AWAY	sec 400

customer 2	BEING SERVED	sec 300

customer 3	IN LINE	sec 200
customer 4	IN LINE	sec 300

customer 5	COMING IN	sec 250

customer 1	TRACE TRACE	800 1200
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
customer 4	TRACE TRACE	1000 -1
customer 5	TRACE TRACE	-1 -1

400 1	MEAN RESPONSE	400

【 3 2 】

sec 1500	FUTURE EVENT	LV 0
sec 2800	FUTURE EVENT	AR 0
sec 3051	FUTURE EVENT	PAUS 0

【 3 3 】

customer 1	TRACE TRACE	800 1200
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
customer 4	TRACE TRACE	1000 -1
customer 5	TRACE TRACE	-1 -1

【 3 4 】

未到者	待ち行列	窓口	退去
5	4, 3	2	1

銀行内			

【 3 5 】

sec 1500	FUTURE EVENT	LV 0
sec 2800	FUTURE EVENT	AR 0
sec 3051	FUTURE EVENT	PAUS 0
sec 0	PAST EVENT	DECL 0
sec 800	PAST EVENT	AR 0
sec 840	PAST EVENT	AR 0
sec 930	PAST EVENT	AR 0
sec 1000	PAST EVENT	AR 0
sec 1200	PAST EVENT	LV 0
sec 1201	PAST EVENT	PAUS 0
customer 1	GONE AWAY	SEC 400
customer 2	BEING SERVED	SEC 300
customer 3	IN LINE	SEC 200
customer 4	IN LINE	SEC 300
customer 5	COMING IN	SEC 250
customer 1	TRACE TRACE	800 1200
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
customer 4	TRACE TRACE	1000 -1
customer 5	TRACE TRACE	-1 -1

400 1	MEAN RESPONSE	400

【 3 7 】

sec 2800	FUTURE EVENT	AR 0
sec 1500	FUTURE EVENT	LV 0
sec 3051	FUTURE EVENT	PAUS 0

sec 800	PAST EVENT	AR 0
sec 840	PAST EVENT	AR 0
sec 930	PAST EVENT	AR 0
sec 1000	PAST EVENT	AR 0
sec 0	PAST EVENT	DECL 0
sec 1200	PAST EVENT	LV 0
sec 1201	PAST EVENT	PAUS 0

customer 5	COMING IN	sec 250

customer 3	IN LINE	sec 200
customer 4	IN LINE	sec 300

customer 2	BEING SERVED	sec 300

customer 1	GONE AWAY	sec 400

customer 5	TRACE TRACE	-1 -1
customer 1	TRACE TRACE	800 1200
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
customer 4	TRACE TRACE	1000 -1

400 1	MEAN RESPONSE	400

【 図 3 8 】

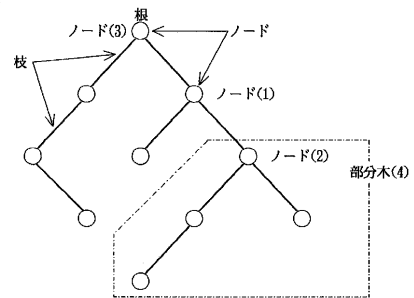
customer 1	GONE AWAY	sec 400
customer 1	TRACE TRACE	800 1200
customer 1	WRITE EVENT	x2 2
customer 2	BEING SERVED	sec 300
customer 2	TRACE TRACE	840 -1
customer 2	WRITE EVENT	x2 2
customer 3	IN LINE	sec 200
customer 3	TRACE TRACE	930 -1
customer 3	WRITE EVENT	x2 2
customer 4	IN LINE	sec 300
customer 4	TRACE TRACE	1000 -1
customer 4	WRITE EVENT	x2 2
customer 5	COMING IN	sec 250
customer 5	TRACE TRACE	-1 -1
customer 5	WRITE EVENT	x2 2
sec 1000	PAST EVENT	AR 0
sec 1000	WRITE EVENT	x2 2
sec 1200	PAST EVENT	LV 0
sec 1200	WRITE EVENT	x2 2
sec 1500	FUTURE EVENT	LV 0
sec 1500	WRITE EVENT	x2 2
sec 2800	FUTURE EVENT	AR 0
sec 2800	WRITE EVENT	x2 2
sec 3051	FUTURE EVENT	PAUS 0
sec 3051	WRITE EVENT	x2 2
sec 0	PAST EVENT	DECL 0
sec 0	WRITE EVENT	x2 2
sec 800	PAST EVENT	AR 0
sec 800	WRITE EVENT	x2 2
sec 840	PAST EVENT	AR 0
sec 840	WRITE EVENT	x2 2
sec 930	PAST EVENT	AR 0
sec 930	WRITE EVENT	x2 2
0 0	WRITE EVENT	x2 2

【 図 3 9 】

sec 2800	FUTURE EVENT	AR 0
sec 800	PAST EVENT	AR 0
sec 840	PAST EVENT	AR 0
sec 930	PAST EVENT	AR 0
sec 1000	PAST EVENT	AR 0
sec 1500	FUTURE EVENT	LV 0
sec 1200	PAST EVENT	LV 0
FUTURE EVENT	DEUS 0	PAUS 0
sec 3051	FUTURE EVENT	PAUS 0
sec 1201	PAST EVENT	PAUS 0
CONTINUE 0	WRITE EVENT	PAUS 0
x0 0	WRITE EVENT	PAUS 0
x1 1	WRITE EVENT	PAUS 0
x2 2	WRITE EVENT	PAUS 0
x3 3	WRITE EVENT	PAUS 0
sec 0	PAST EVENT	DECL 0
customer 1	TRACE TRACE	800 1200
customer 4	TRACE TRACE	800 -1
customer 2	TRACE TRACE	840 -1
customer 3	TRACE TRACE	930 -1
customer 5	TRACE TRACE	-1 -1

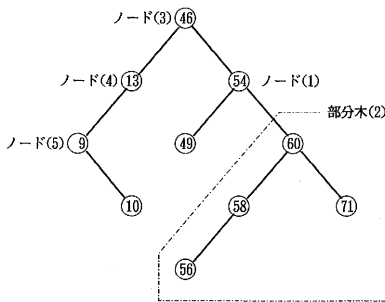
【 図 4 0 】

二分木



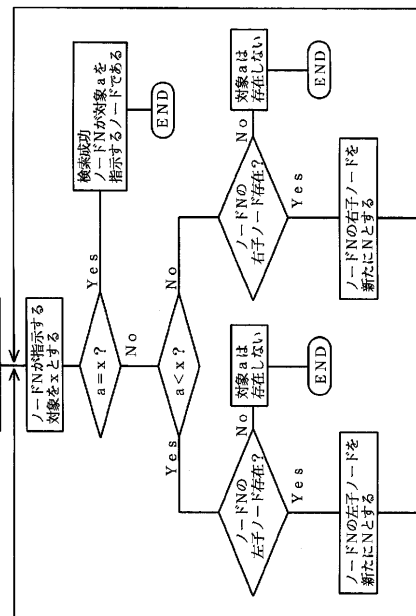
【 図 4 1 】

二分探索木



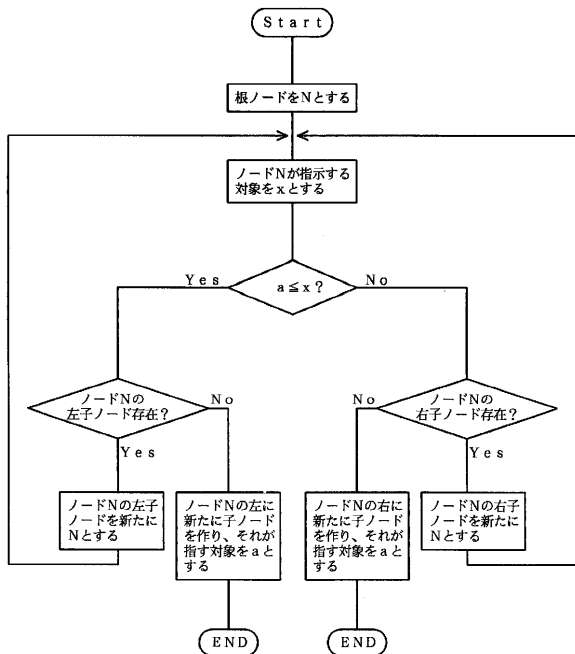
【 図 4 2 】

【 二分探索木での対象 a の探索 】



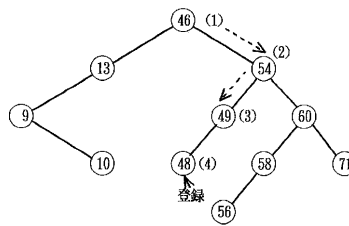
【 図 4 3 】

[二分探索木での対象 a の登録]



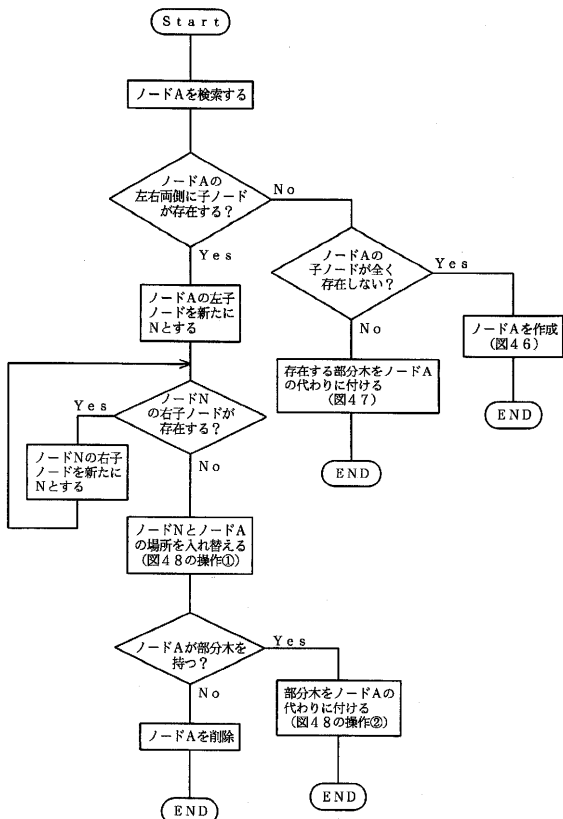
【 図 4 4 】

[整数 48 の登録]

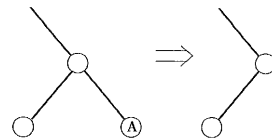


【 図 4 5 】

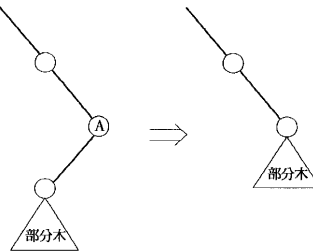
[二分探索木での対象 a を指示するノード A の削除]



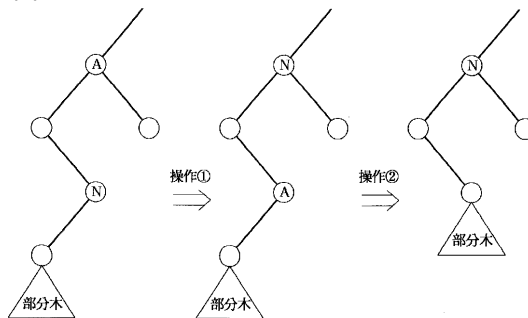
【 図 4 6 】



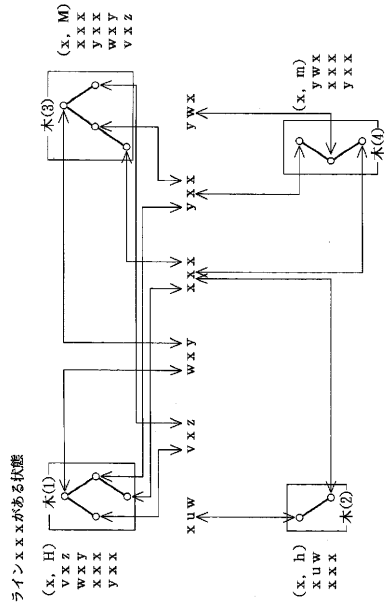
【 図 4 7 】



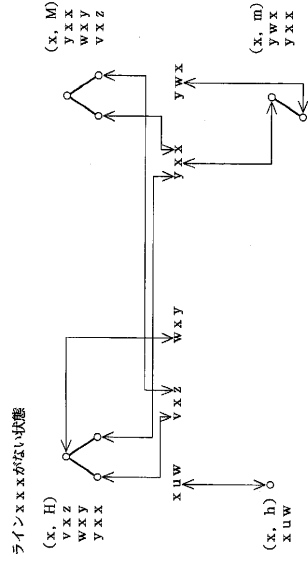
【 図 4 8 】



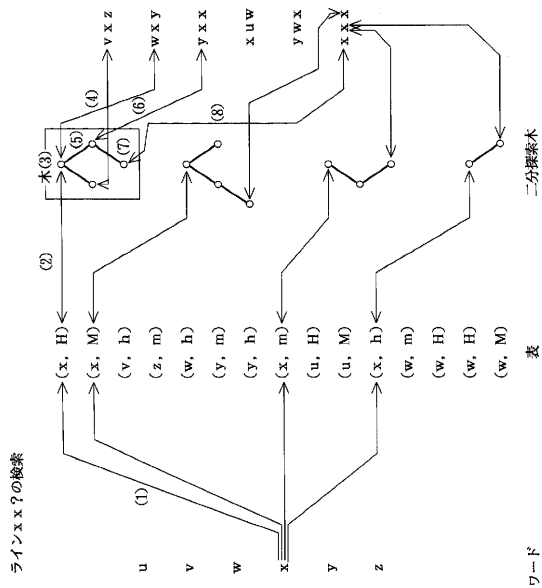
【 図 49 】



【 図 50 】



【 図 51 】



【 図 53 】

従業員リレーション

番号	氏名	生年	職種	都
1	上村 誠	' 61	工員	製造
2	斎藤 二郎	' 67	販売	営業
3	三浦 浩二	' 63	工員	製造
4	飯田 康夫	' 65	工員	製造
5	藤原 清	' 64	販売	営業

部	部長
製造	田中 一郎
営業	佐々木洋二

【 図 52 】

都県-市リレーション

都県	市
東京	三鷹
東京	武蔵野
神奈川	川崎
神奈川	横浜

フロントページの続き

- (72)発明者 今野 宏
神奈川県川崎市幸区柳町70番地 株式会社東芝 柳町工場内
- (72)発明者 和田 維作
神奈川県川崎市幸区柳町70番地 株式会社東芝 柳町工場内

審査官 土田 行一

- (56)参考文献 特開平04-219871(JP,A)
疋田 定幸, 図解分散型データベースシステム入門, 日本, 株式会社オーム社, 1989年 5月25日, 第1版, P.43-50, 分散型データベースの分割に対する透過性について例示するものである
- (58)調査した分野(Int.Cl.⁷, DB名)
G06F 12/00