

テーブル設計の基本

平成 19 年 1 月 30 日

(有)ビット

浦 秀樹

ドキュメント『超簡単売上入力フォームの作成』では、テーブル設計についてそれほど詳しくは解説しなかった。正直なところ、あの程度のテーブル設計ができない人は Access など使えない、そう思っていたからだ。仮に、データベース初心者であってもあの参考プログラムを見れば何かを感じてくれるだろう、きっと勉強になるはずだとも思った。

しかし、よく考えるとデータベース初心者になれば、なぜあのように一对多の関係のテーブル設計にしなければならないのか理解できない、ひょっとしたらそのあたりのことが非常に重要ではなかろうかと思いはじめた。そこで、あらためてこちらでリレーショナルデータベースシステムにおけるテーブル設計の基本について解説することにした。

もしも Access ではなく、EXCEL であのような売上入力管理を行って欲しいといわれたら、多くの人は次のようなワークシートを作成するだろう。

[表 1]

伝票番号	売上日	得意先コード	得意先名	商品コード	商品名	数量	単価	金額
1000	07/01/01	2000	浦商店	1001	商品 1	1	100	100
1000	07/01/01	2000	浦商店	1010	商品 10	1	500	500
1000	07/01/01	2000	浦商店	2050	商品 250	3	800	2400
1001	07/01/15	1800	河北商事	1002	商品 2	2	150	300
1001	07/01/15	1800	河北商事	1001	商品 1	6	100	600

この表は決して間違いではない。Access であったとしてもこのようなテーブルを用いて入力画面を作ることとは不可能ではないし、売上情報を記録するには必要十分だ。

ところがこの表を用いて何百枚もの売上傳票を入力するとしたらどうだろうか。実際に入力してみると様々な問題が見えてくるはずだ。

たとえば、[伝票番号]、[売上日]、[得意先コード]、[得意先名]、[商品コード]、[商品名]などは同じことを何度も繰り返して入力しなければならない。これを何とかしたいと思うのは当然のこと。そして、EXCEL 上級者はすぐに得意先や商品の一覧を別のワークシートに登録し、LOOKUP 関数などで参照できるようにするはずだ。実はその発想こそがリレーショナルデータベースシステムに踏み込む第一歩なのだ。

※[得意先名]や[商品名]のように、何度も繰り返して入力するデータを別のワークシートやテーブルに分けて利用することを『正規化』という。

だが、仮に LOOKUP 関数を使って得意先名の入力が楽になったとしても、[伝票番号]や[得意先コード]は明細数のぶんだけ入力しなければならないことには変わりはない。このことに多くの人は素直に疑問を感じるだろう。

その疑問を解決すべく次のような2つのテーブルで同じ情報を入力したらどうだろうか？

伝票番号	売上日	得意先コード	得意先名
1000	07/01/01	2000	浦商店
1001	07/01/15	1800	河北商事

[表 2-1]

伝票番号	商品コード	商品名	数量	単価	金額
1000	1001	商品 1	1	100	100
1000	1010	商品 10	1	500	500
1000	2050	商品 250	3	800	2400
1001	1002	商品 2	2	150	300
1001	1001	商品 1	6	100	600

[表 2-2]

これら2つのテーブルでデータ入力を行ってみると、確かに重複データは少なくなった（伝票番号が重複しているが）。重複するデータが少なくなれば入力操作もスピーディーに行われるし、データを記憶するメモリー資源も少なくてすむ。

しかし、情報が二つのテーブルに分かれるので面倒なことも出てくる。まず第一にデータ入力時に二つのテーブルを交互に渡り歩かねばならないということ。ただし、これは売上入力のようなフォームを活用することで改善はできる。（実際には Access などではテーブルを直接開いてダイレクトにデータ入力するということはほとんど無い。仮に、そのような使い方をするのであれば EXCEL の方がよっぽど使いやすい。）

問題となるのは入力時ではなく、出来上がったデータをいろいろと活用するときに発生するのだ。たとえば、上記のデータを使って得意先ごとの売上集計を行うとしたら、どうしてもこの二つのテーブルを一つの表に結合させて扱いたくなるはずだ。実は、その結合作業を受け持つのがクエリなのだ。

すでにお分かりかもしれないが、リレーショナルデータベースシステムでは上のような売上データを二つのテーブルに分けて利用することが多い。

この二つのテーブルの関係は『[伝票番号]でリンクしている』と表現される。しかし、間違っているのではないのは、『リンクしている』のはテーブルとテーブルの関係であって、この段階ではあくまでも作り手の頭の中だけのこと。この時点では Access はなんの特別扱いもしてくれないということだ。※特別扱いするのは、後述する『リレーションシップ』の設定をおこなったときからだ。

[表 1]では[伝票番号]は主キーとして設定される。主キーとは、テーブルの中の唯一のレコードを呼び出すことができる目印となるフィールドのことだ。主キーとなったフィールドの値は、テーブル内に二つと同じものは存在しないというルールとなる。

そしてこのような主キーを有するテーブルと、関連するもう一つのテーブルの結びつきを『一対多の関係、または結合』という。

ではここであらためて Access で作成された売上入力画面を見てみよう。

売上メイン

新規入力 削除 行挿入 印刷 閉じる

伝票番号: 1007 売上日: 2007/01/12 担当者: 10 王

得意先: 100 山田太郎 金額合計: 149,600

商品コード	商品名	数量	単位	単価	金額
▶ 1000	パソコンA	1.	台	10,000.	10,000.
1001	パソコンB	10.	台	12,000.	120,000.
1002	パソコンC	2.	台	9,800.	19,600.
米		0.		0.	

レコード: 1 / 6

フォームを見れば分かるように、このフォームで表示されているデータこそがすでに『一对多』だ。だから、素直に必要なとされるテーブルも一对多となるのは極々自然な成り行きなのである。Access では真ん中のフォームをサブフォーム、外側のフォームをメインフォームと言う。

フォームを作成する段階で、二つのテーブルはメインフォームとサブフォームとに分かれて作成され、サブフォーム定義時にサブフォームのどのフィールドとメインフォームのどのフィールドとリンクするかを決める。もちろん、この場合は[伝票番号]でリンク設定するということになる。リンクさせるフィールド名は必ずしも同じ名前ではないというわけではない。

リンク設定は、サブフォームオブジェクトのプロパティを開き、[リンク子フィールド]、[リンク親フィールド]で設定される。

リンクフィールドをサブフォーム側に設定すると、指定されたフィールドの値は入力時に自動的にメインフォームからサブフォームへと複写される。

サブフォーム/サブレポート: 売上サブのサブフォーム

売上サブのサブフォーム

書式 データ イベント その他 すべての

ソースオブジェクト: 売上サブのサブフォーム

リンク子フィールド: 伝票番号

リンク親フィールド: 伝票番号

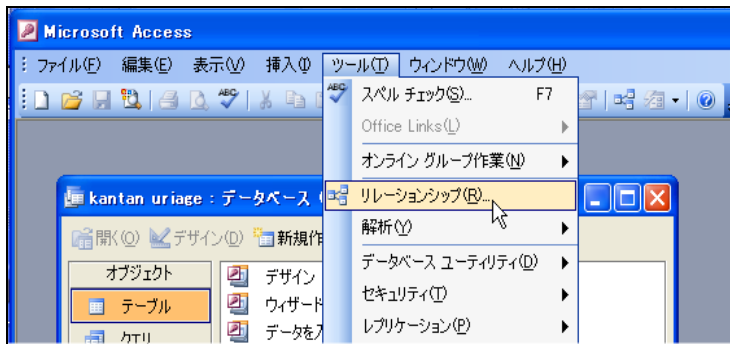
使用可能: はい

編集ロック: いいえ

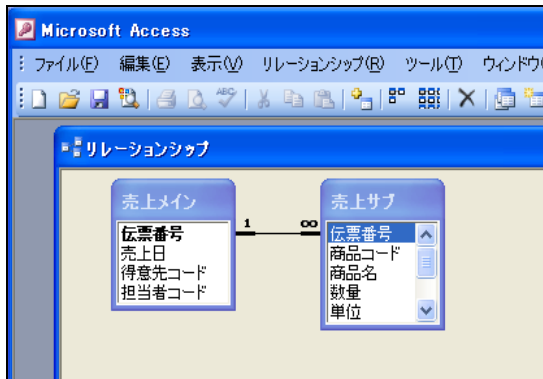
Access ではこれら二つのテーブルの関係を次に示す『リレーションシップ』という設定でさらに厳格な結びつきへと設定することができる。しかし、このような一对多を表すフォームで、対象となるテーブル等がリレーションシップ設定されていないというわけではない。上図のごとく、フォーム上でリンク設定さえ行われていれば、フォーム上のメインサブ関連動作は問題ない。

再度述べるが、フォーム作成時に設定するリンク設定と、テーブルのリレーションシップ設定とは別物である。混同しないよう要注意である。

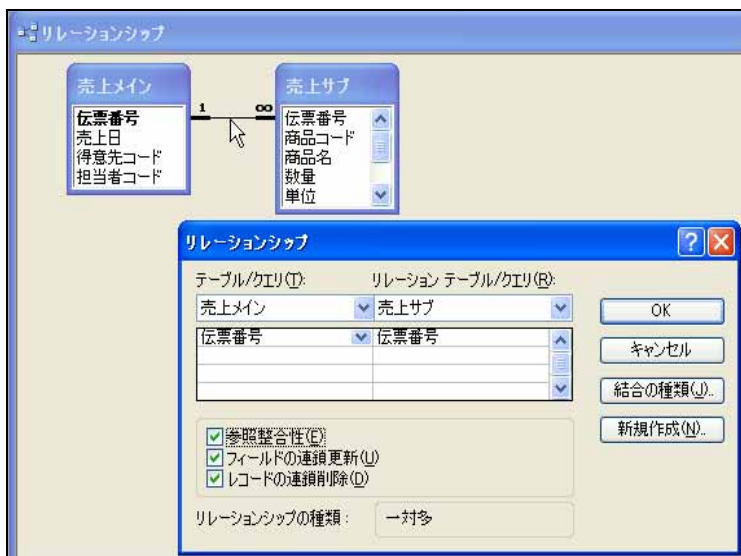
リレーションシップの設定は、メニュー[ツール]、[リレーションシップ]をクリックして行なわれる。



※メニュー[ツール]、[リレーションシップ]をクリックして設定画面を表示する。



※上の図、[売上メイン]テーブルと[売上サブ]テーブルのリレーションシップ設定は次の通り。



参照整合性、フィールドの連鎖更新、レコードの連鎖削除の詳細は次の通り。

項目	解説
参照整合性	このチェックをオンにすると関連するテーブルの整合性が保障される。たとえば、一（一対多の側のこと）側テーブル側に存在しない[伝票番号]のレコードを多側に作成することはできなくなる。多側テーブルを直接開いて変更しようとしても変更後の値が一側に存在しないと拒否される。
フィールドの連鎖更新	ここをオンにすると、たとえば一側の[伝票番号]を変更すると、多側のテーブルの[伝票番号]が自動的にその値に置き換えられる。
レコードの連鎖削除	ここをオンにすると、一側を削除した場合に多側の関連レコードが自動的に削除され

売上入力用のテーブルではリレーションシップの画面ですべて項目にチェックを入れることが多い。

さて、売上入力画面には見逃してはならない重要なフィールドが2つある。売上明細の[金額]フィールドとその[金額]の合計を表す[金額合計]フィールドがそれだ。

実は、これらはフォームやクエリに仕掛けられた計算式で求められるものであって、決してテーブルでその値を保持しているのではない。

リレーショナルデータベースシステムにおいて、計算で求められる値をフィールドの値として持たせないことが原則だ。

ただし、仮にそのようなフィールドを設定したとしても間違いではない。場合によってはそのほうが処理レスポンスに優れる場合もある。しかし、プログラミング上、様々な問題をはらんでくのをあらかじめ知っておいたほうがよい。

●テーブル（またはクエリ）とフォームの関係

Accessにとってテーブルは単なるデータの入れ物に過ぎない。フィールド同士の計算もできないし、列の合計値も算出できない。ただし、データベース管理システムにおいてのこの入れ物は表計算ソフトのように何でもかんでも受付けるのではない。そこに入れるデータの種類や特徴を厳しく限定してくれる。そのことによってデータベースの利用者は安心して利用できることになる。そして、データはフォームやレポートを通して様々な形で表現される。分散されたデータを共通のフィールドで結合させて一つのテーブルとして表示したり、抽出結果を表示したり、計算結果を表示したり・・・。

●インデックスについて

インデックスは書籍などと言うところの目次にあたるものだ。データを探し出すときに、すべてのページを順番に探すよりも、あらかじめ目次があればそれを利用すると検索処理速度は飛躍的に高速となる。

たとえば、ある伝票番号のデータを探し出そうとする際に、Accessのシステムはインデックスデータの中で目的のデータを探し出す。目的のデータが見つかり、そこにはハードディスクの何丁目何番地に目的のデータが記録されているという情報があり、それによってシステムはいっきに目的の場所をシークし始める。

インデックスはリンクに関するフィールドだけではなく、検索に利用されるすべてのフィールドに設定する。たとえば、売上入力テーブルの多側の明細テーブルにおいて、[伝票番号]には必ずインデックスを設定しなければならない。そうすることによって、クエリ実行時に処理が高速となるわけだ。ほかにも、一側の[売上日]や[得意先コード]、多側の[商品コード]や[商品名]などにもインデックスを設定するとよいだろう。

また、インデックスを設定するときに[重複禁止]とすることでさらに検索処理が高速化される。主キーに設定されたフィールドは自動的にインデックスが設定され、さらに重複禁止設定となる。

以上