

## 超簡単、売上入力フォームの作成

平成 19 年 1 月 10 日  
有限会社ビット  
浦 秀樹

はじめに

ビットの解説ドキュメントをダウンロード下さいまして誠にありがとうございます。  
このドキュメントは、Access をより活用したいとお考えの皆様へ書き上げました。  
ドキュメントの内容に沿ってプログラムを体験するだけで、知らず知らずに Access が修得できる、そういったものを目指しました。

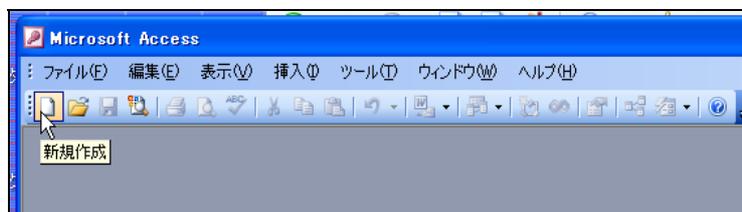
商品コード	商品名	数量	単位	単価	金額
▶ 1000	パソコンA	1.	台	10,000.	10,000.
1001	パソコンB	10.	台	12,000.	120,000.
1002	パソコンC	2.	台	9,800.	19,600.
*		0.		0.	

これから作成しようとする売上入力フォーム

### データベースの作成

最初にデータベースの作成を行う。データベースの作成手順は次の通り。

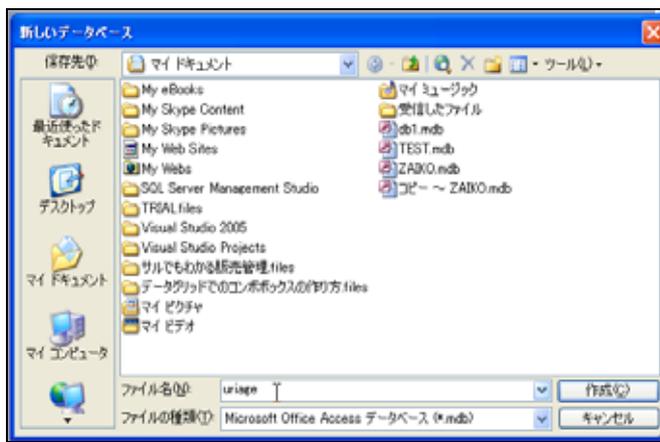
1. 新しいデータベースを作成するには Access ウィンドウの[新規作成]アイコンをクリックする。



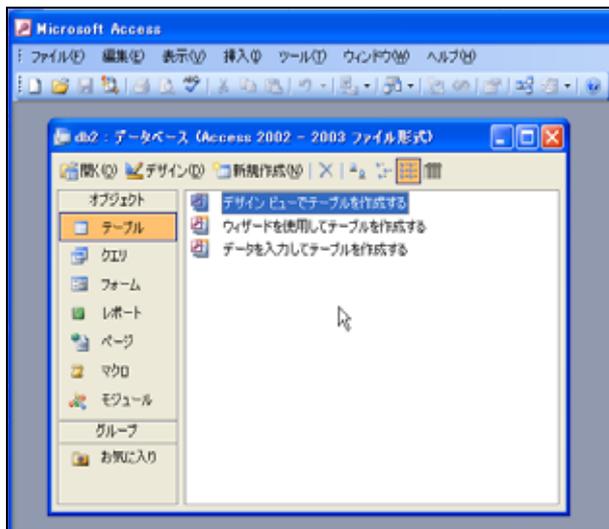
2. Access ウィンドウ右横の[空のデータベース]をクリックする。



3. [新しいデータベース]画面でファイル名を任意に付けて[作成]ボタンをクリックすると



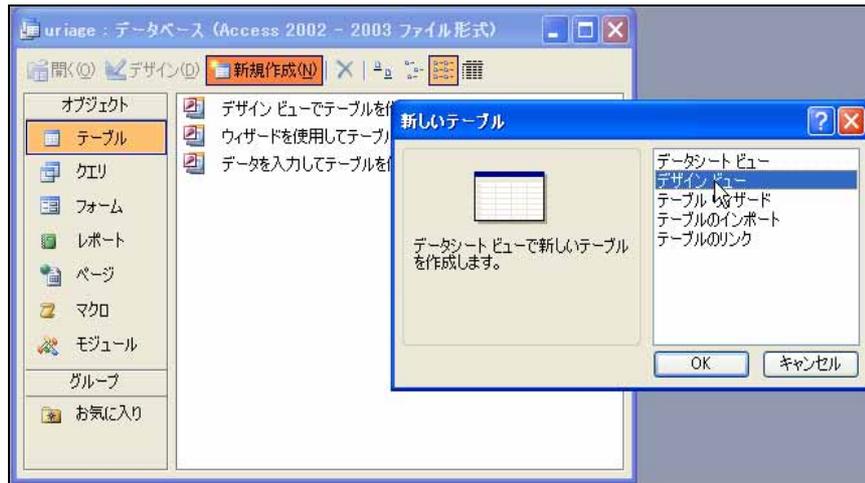
4. 作成されたデータベースウィンドウが表示される。  
これでデータベースの作成が完了。



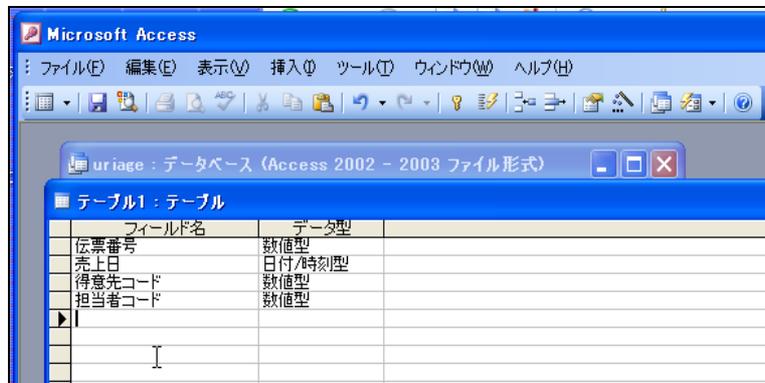
## テーブル設計

次にテーブル設計だが、テーブルを設計する前に売上入力画面を手書きでもかまわないから下書きしてみよう。この下書きが書けないのならテーブル設計は不可能だ。  
これから作成しようとする売上入力画面のイメージがはっきりしたらテーブル作成に取り掛かる。

1. データベースウィンドウで[テーブル]オブジェクトを選択して、[新規作成]ボタンをクリックすると[新しいテーブル]ダイアログが表示される。  
[デザインビュー]を選択して[OK]ボタンをクリックする。



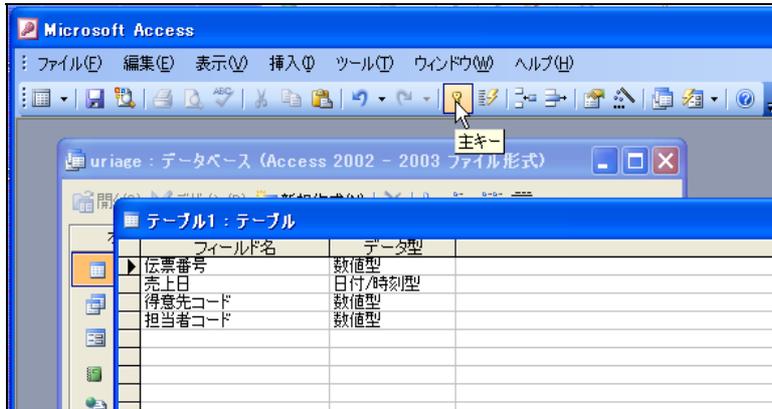
2. テーブルのデザイン画面で次のようにフィールドを登録する。



[売上メイン]テーブル定義情報

フィールド名	データ型
伝票番号	数値型 (長整数)
売上日	日付 / 時刻型
得意先コード	数値型 (長整数)
担当者コード	数値型 (長整数)

3. 作成したテーブルの[伝票番号]を主キーに設定する。  
 主キーとは、テーブルのデータを呼び出す際に利用される重要なもので、『同じものが二つと無い』(重複が禁止される)ことを示すものである。データの呼び出しにはこの主キーを使って行われることが多い。



4. 同様に[売上サブ]テーブルを次のように作成する。今回のこのテーブルには主キーは必要ない。

売上サブ: テーブル	
フィールド名	データ型
伝票番号	数値型
商品コード	テキスト型
商品名	テキスト型
数量	通貨型
単位	テキスト型
単価	通貨型
枝番	数値型

[売上サブ]テーブル定義情報

フィールド名	データ型
伝票番号	数値型 (長整数)
商品コード	テキスト型
商品名	テキスト型
数量	数値型 (倍精度)
単位	テキスト型
単価	通貨型
枝番	数値型 (倍精度)

5. 続いて、担当者マスタ、商品マスタ、得意先マスタを次のように作成する。  
 印は主キーを意味する

[担当者マスタ]テーブル定義情報

フィールド名	データ型
担当者コード	数値型 (長整数)
担当者名	テキスト型

[商品マスタ]テーブル定義情報

フィールド名	データ型
商品コード	テキスト型
商品名	テキスト型
単位	テキスト型
単価	通貨型
課税	Yes/No 型

[得意先マスタ]定義情報

フィールド名	データ型
得意先コード	数値型 (長整数)
得意先名	テキスト型
よみ	テキスト型

6. 各テーブルにデータを入力する。

担当者コード	担当者名
10	王
11	長島
12	金田
13	柴田

商品コード	商品名	単位	単価	課税
1000	パソコンA	台	¥10,000	<input type="checkbox"/>
1001	パソコンB	台	¥12,000	<input type="checkbox"/>
1002	パソコンC	台	¥9,800	<input type="checkbox"/>
2000	ディスプレイ1	台	¥3,000	<input type="checkbox"/>
2001	ディスプレイ2	台	¥3,500	<input type="checkbox"/>
3000	ハードディスク1	台	¥5,000	<input type="checkbox"/>
3001	ハードディスク2	台	¥6,000	<input type="checkbox"/>
3002	ハードディスク3	台	¥7,000	<input type="checkbox"/>
			¥0	<input type="checkbox"/>

得意先コード	得意先名	よみ
100	山田太郎	やまだたろう
101	上島二郎	うえしまじろう
102	鈴木一郎	すずきいちろう
103	浦秀樹	うらひでき
104	長田真一	ながたしんいち
105	吉田幸子	よしたゆきこ
106	伊藤花子	いとうはなこ
*	0	

## フォーム設計

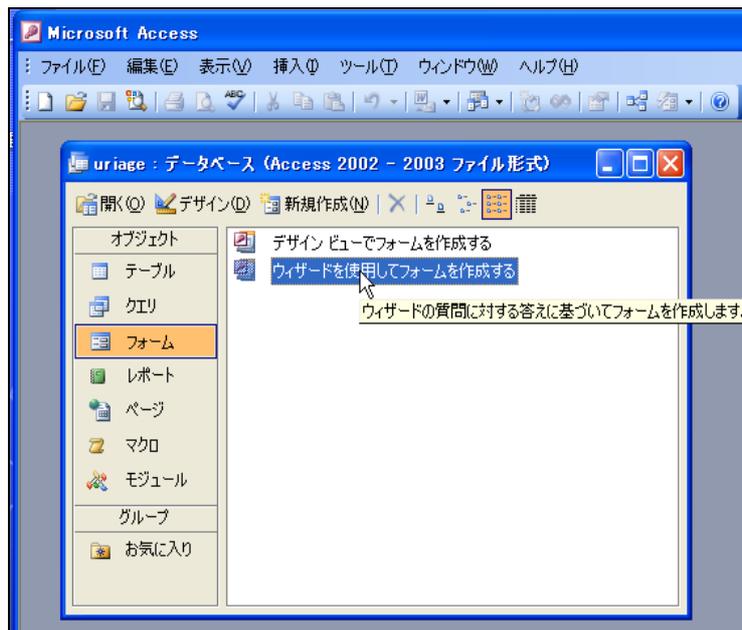
テーブルが完成したらいよいよ売上入力フォームの作成だ。

ここでは、フォームの中に別フォーム（サブフォーム）を表示する、リレーショナルデータベースとしては利用価値の高いフォームを作成する。

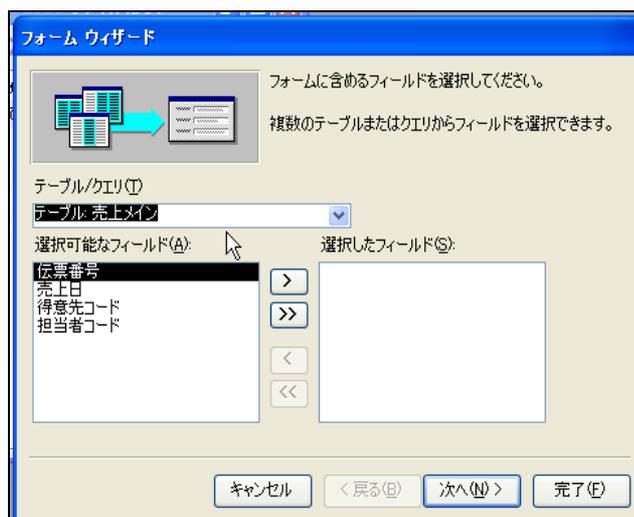
大まかな手順としては、最初にメインフォームを作成し、次の中で表示されるサブフォームを作成する。サブフォームを先に作成するという方法もある。

作成手順は次の通り。

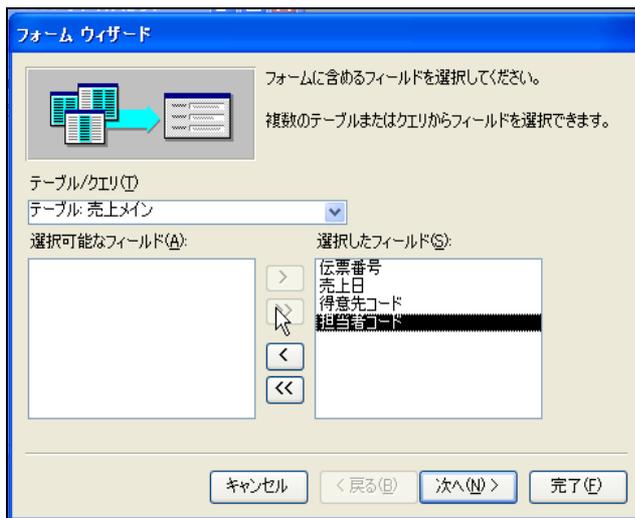
1. データベースウィンドウの[フォーム]を選択して[ウィザードを使用してフォームを作成する]をクリックする。



2. フォームウィザードで[売上メイン]テーブルを指定する。



- 表示された[売上メイン]テーブルの全てのフィールドを選択フィールド(画面右側)に移動させる。[次へ]ボタンをクリックする。



- [単票形式]を選択して[次へ]ボタンをクリックする。



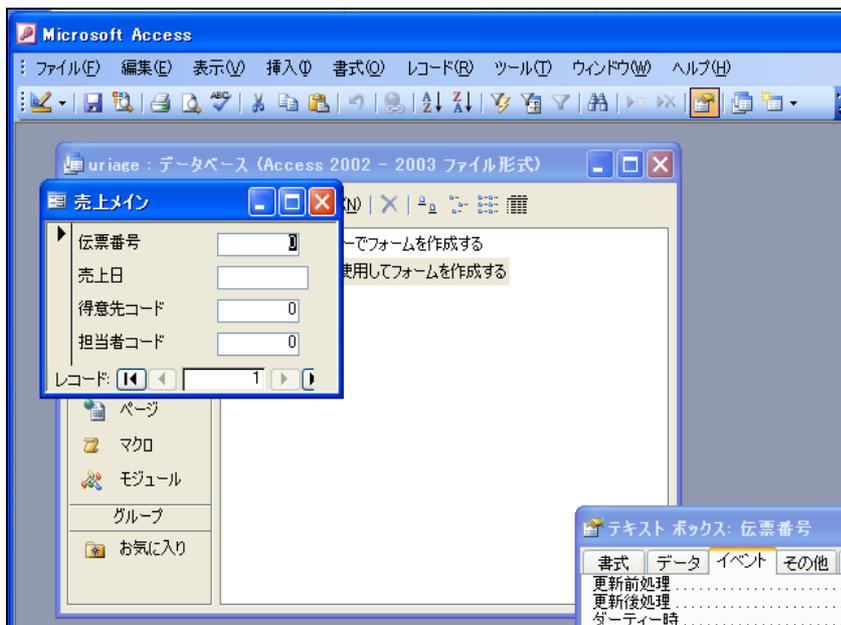
- スタイルを[標準]としたままで[次へ]ボタンをクリックする。



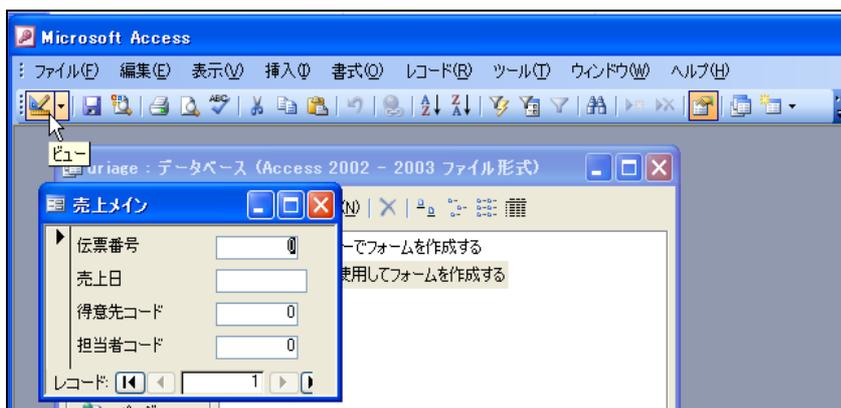
6. フォーム名を確認して[完了]ボタンをクリックする。



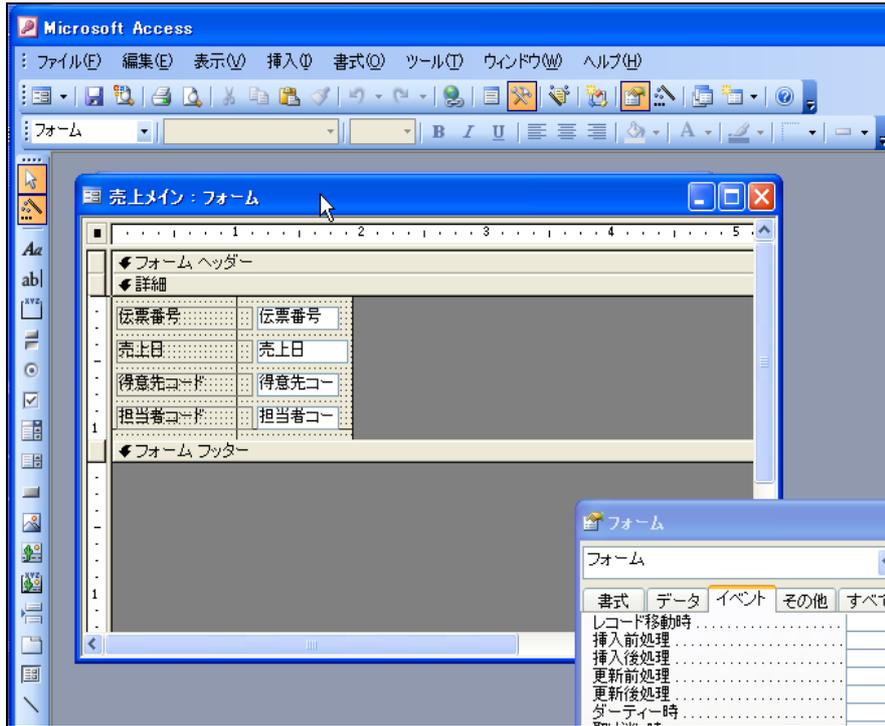
7. ウィザードで作成された[売上メイン]フォームが表示される。このときはまだ図のように小さいフォームとなっている。



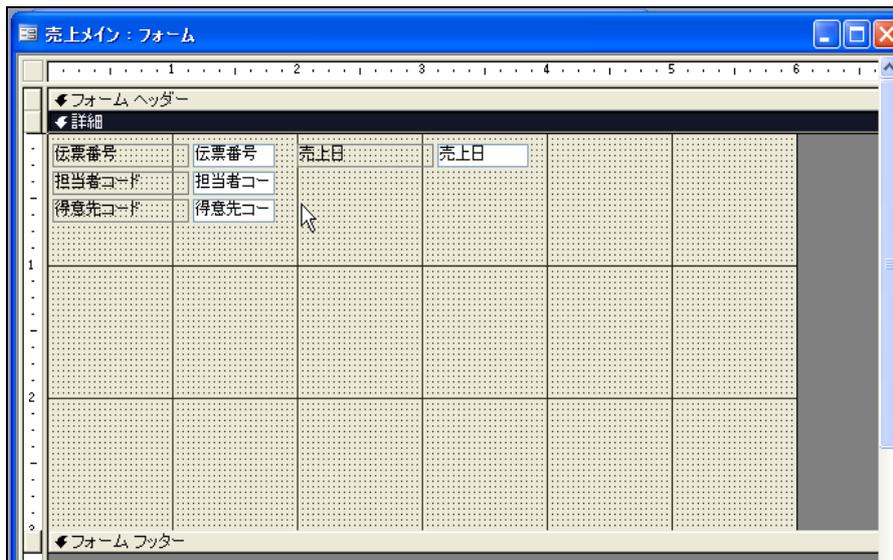
8. フォームを修正するために、[ビュー]アイコンをクリックしてフォームデザインビューに切り替える。



9. フォームのデザインビューが表示される。

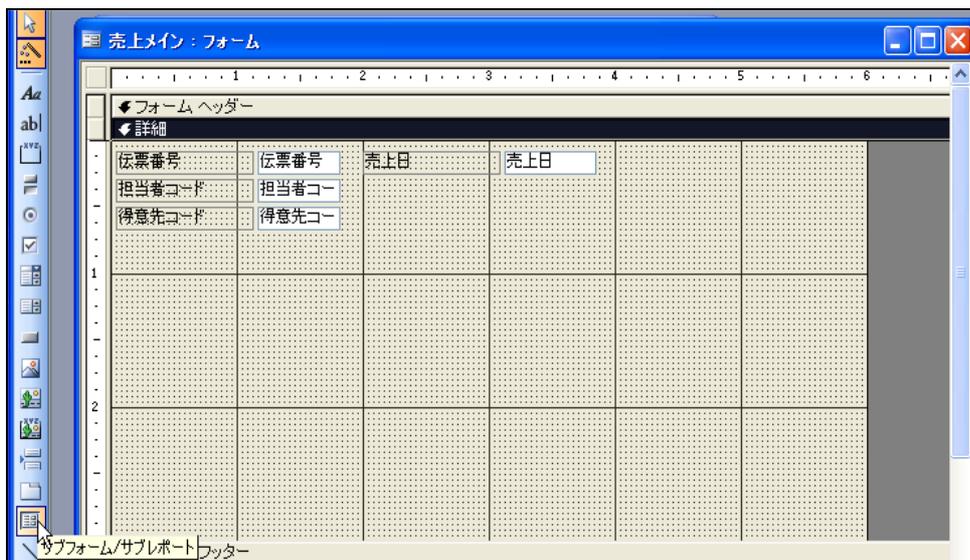


10. フォームの大きさを変更し、テキストボックスの位置を任意に変更する。

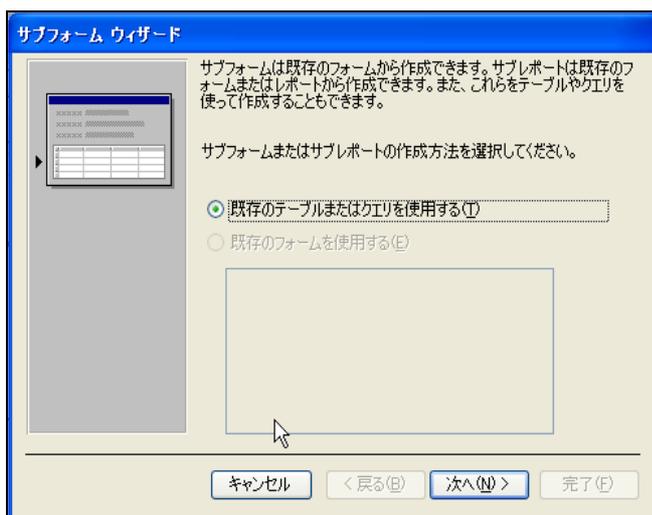


11. サブフォームを作成するために、デザインビュー左にある[サブフォーム/サブレポート]アイコンをクリックする。

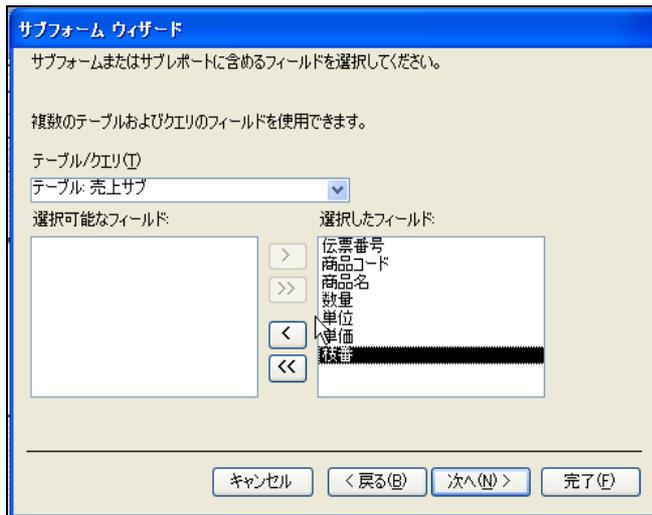
このアイコンが見当たらないときは、メニュー[表示]、[ツールボックス]をクリックすると表示される。



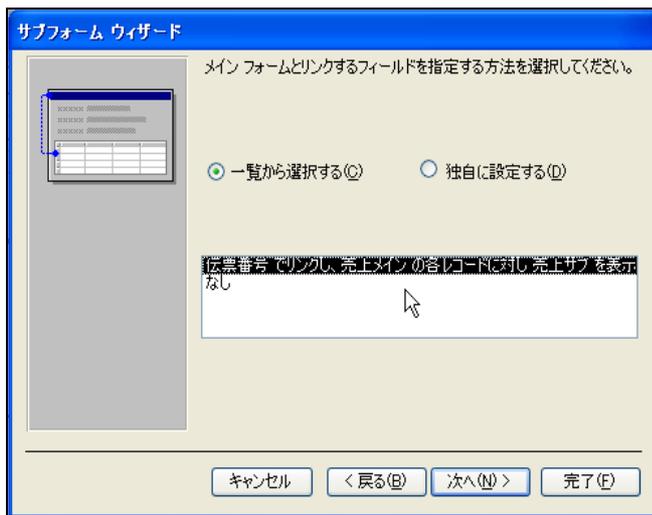
12. サブフォームウィザードが表示されたら、[次へ]ボタンをクリックする。



13. テーブル/クエリですでに登録されている[売上サブ]テーブルを指定して、すべてのフィールドを[選択したフィールド]に移動させる。



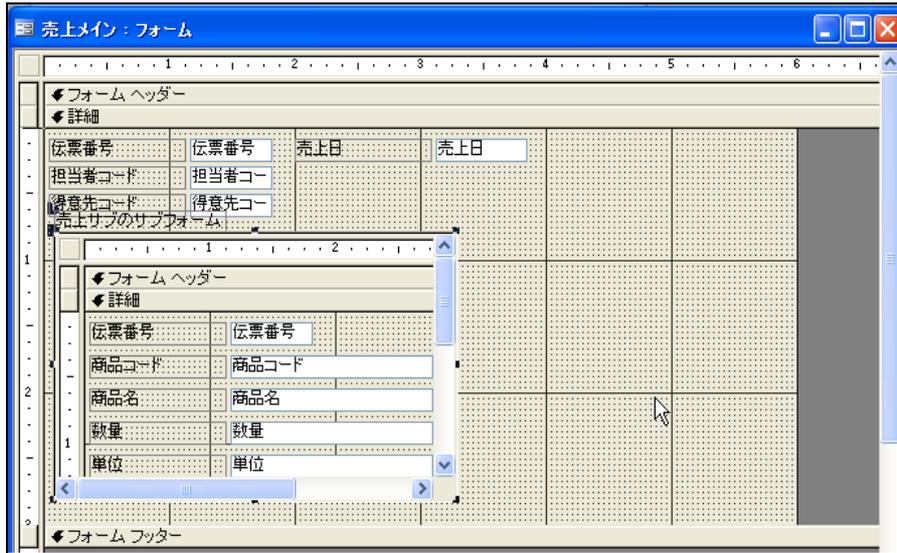
14. メインフォームとサブフォームがリンクする方法を選択する。この場合は、[伝票番号でリンク...]を選択し、[次へ]ボタンをクリックする。



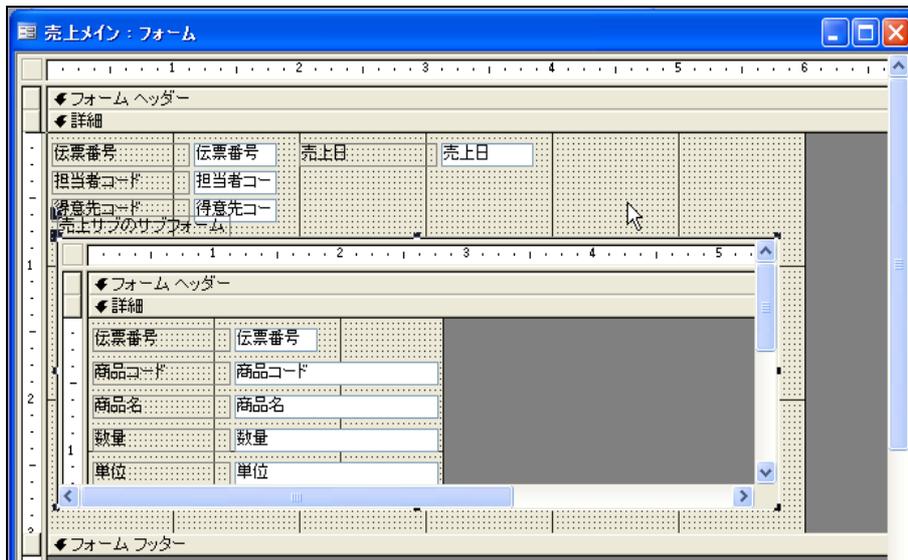
15. サブフォーム名を確認して、[完了]ボタンをクリックする。



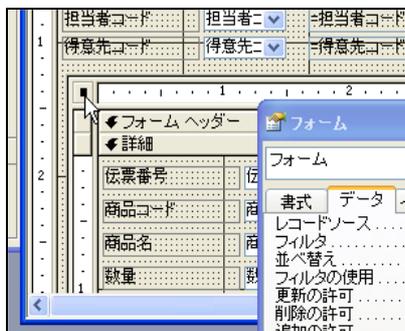
16. サブフォームが自動的に作成され、図のようなフォームとなる。



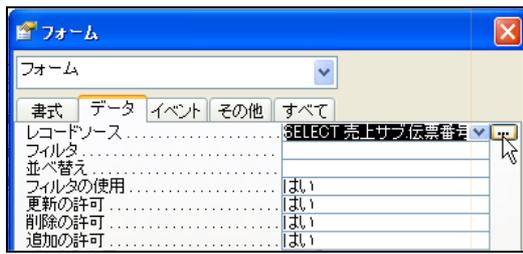
17. サブフォームを任意の大きさに変更する。



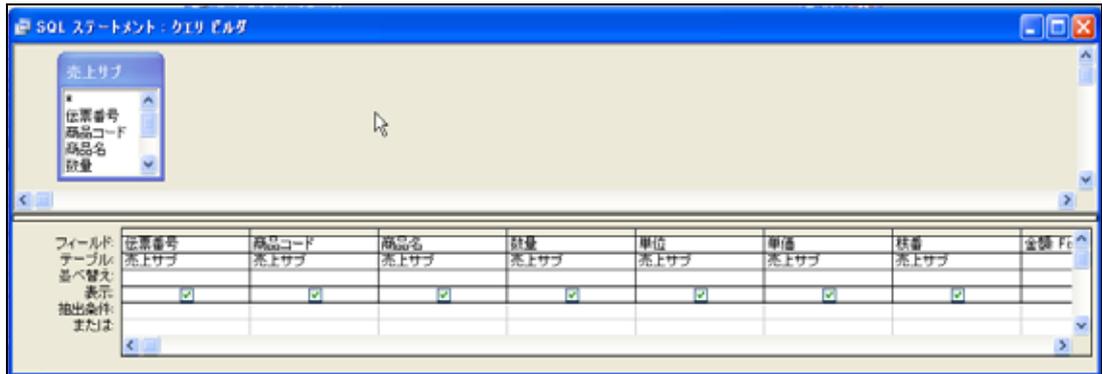
18. この状態ではサブフォーム上のデータの並び順が決まっていないので、サブフォームのレコードソースを開いてクエリの[並べ替え]を指定する。  
レコードソースのクエリを表示するためには、サブフォームの左上隅をダブルクリックする。



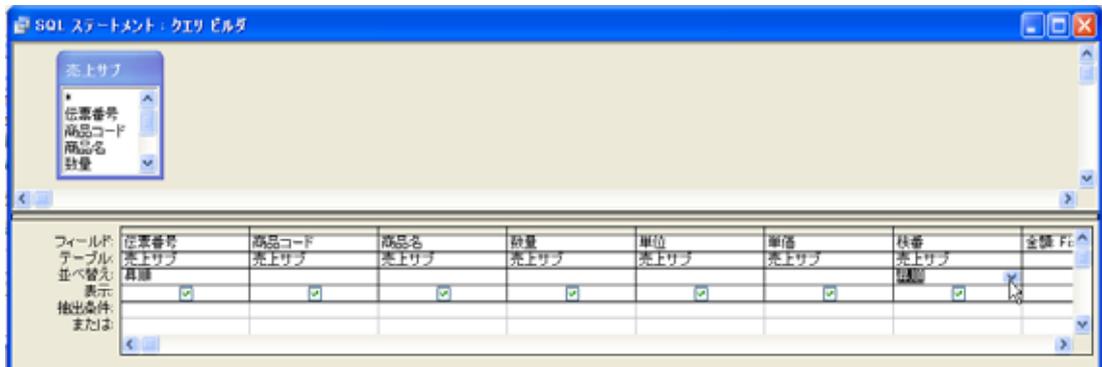
19. 表示されたプロパティシートの[データ]タブをクリックして、[レコードソース]の右端アイコンをクリックする。



20. クエリビルダ画面が表示される。

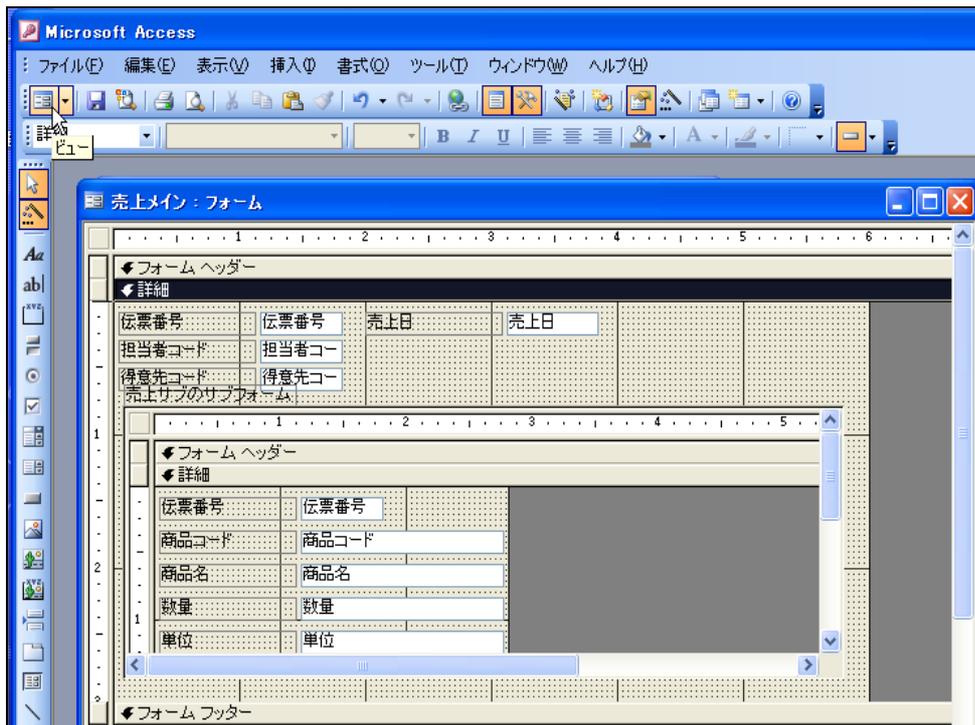


21. フィールド欄で[伝票番号]フィールドと[枝番]フィールドに[並べ替え]の設定を『昇順』としてビルダを閉じる。

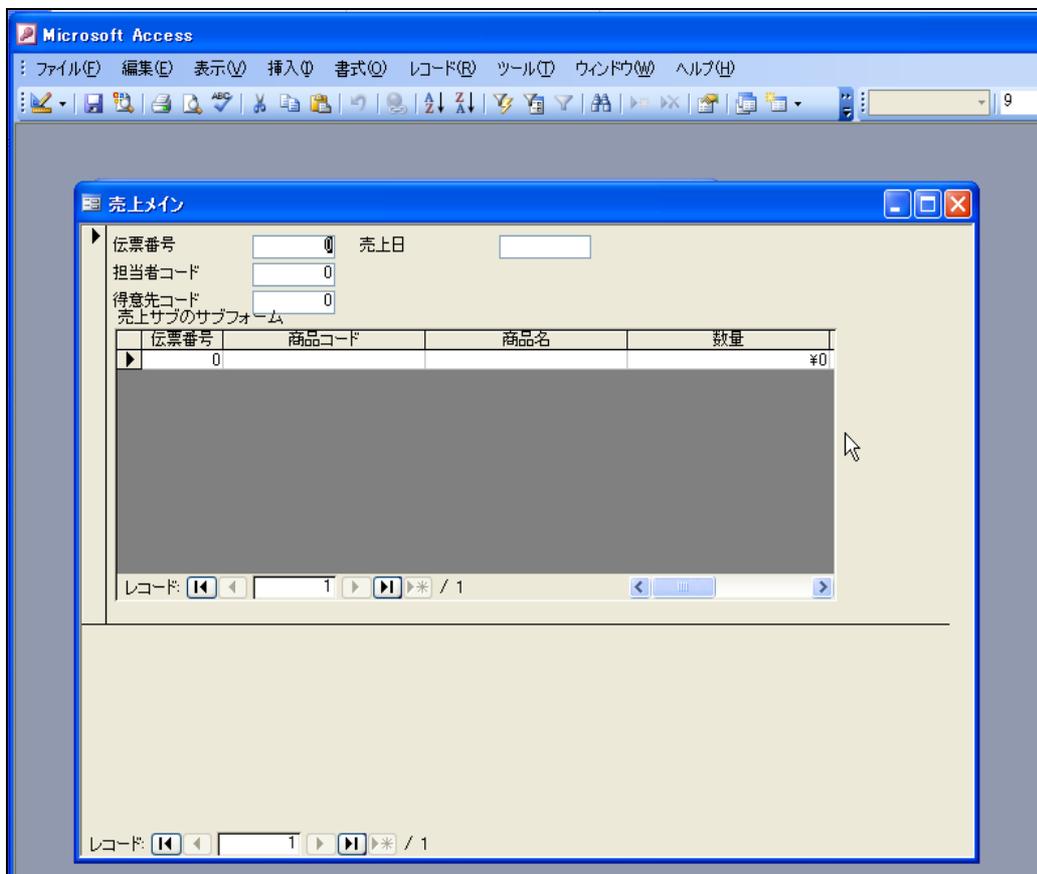


これにより、サブフォーム上では[伝票番号]ごとの[枝番]順にデータが並び変わる。

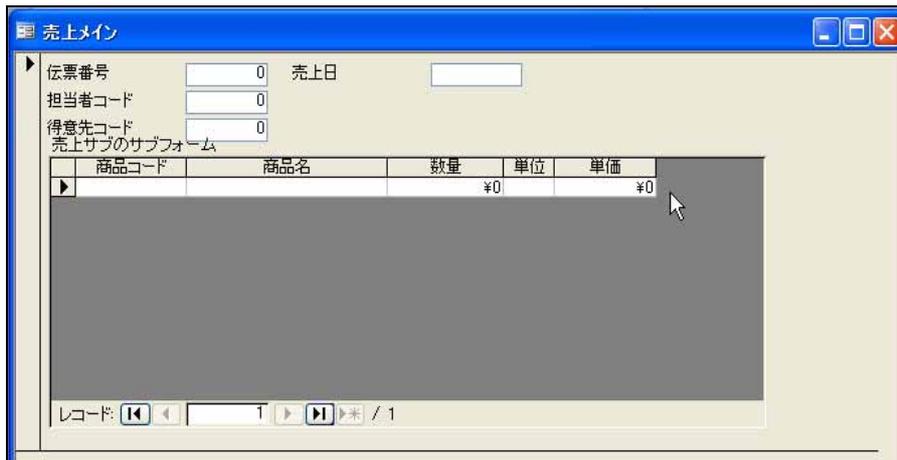
22. ここで一旦フォームをデザインビューからフォームビューに切り替えて見た目を確認してみよう。確認するには[ビュー]アイコンをクリックしてフォームビューに切り替える。



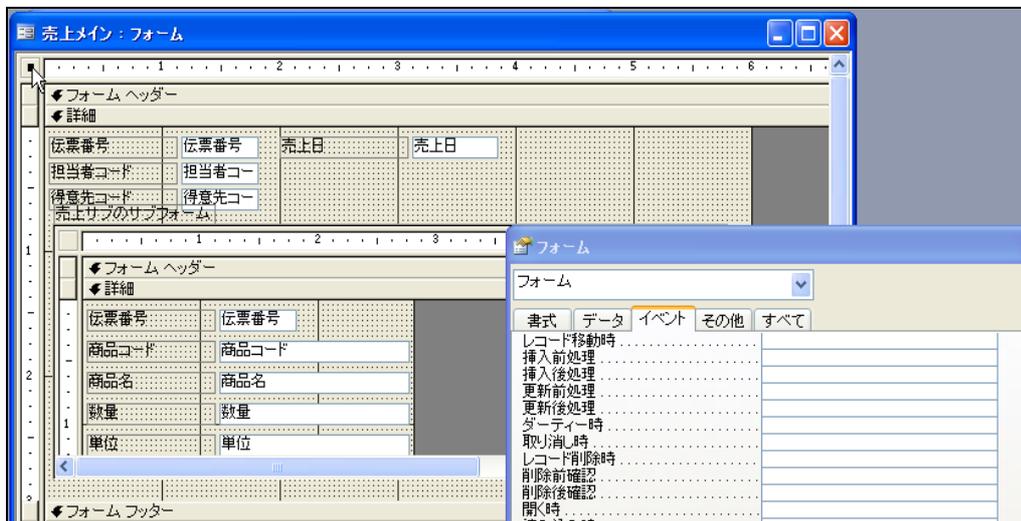
23. フォームが表示された。このときはまだサブフォームの中に必要の無いフィールドが表示されている。



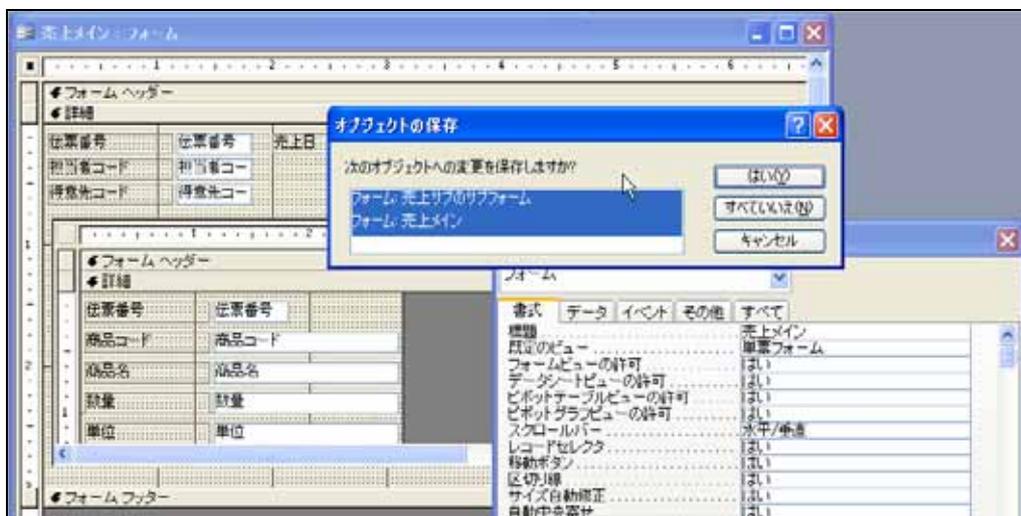
24. サブフォームのフィールド境界線をマウスでドラッグして unnecessary フィールドを画面上から消す。



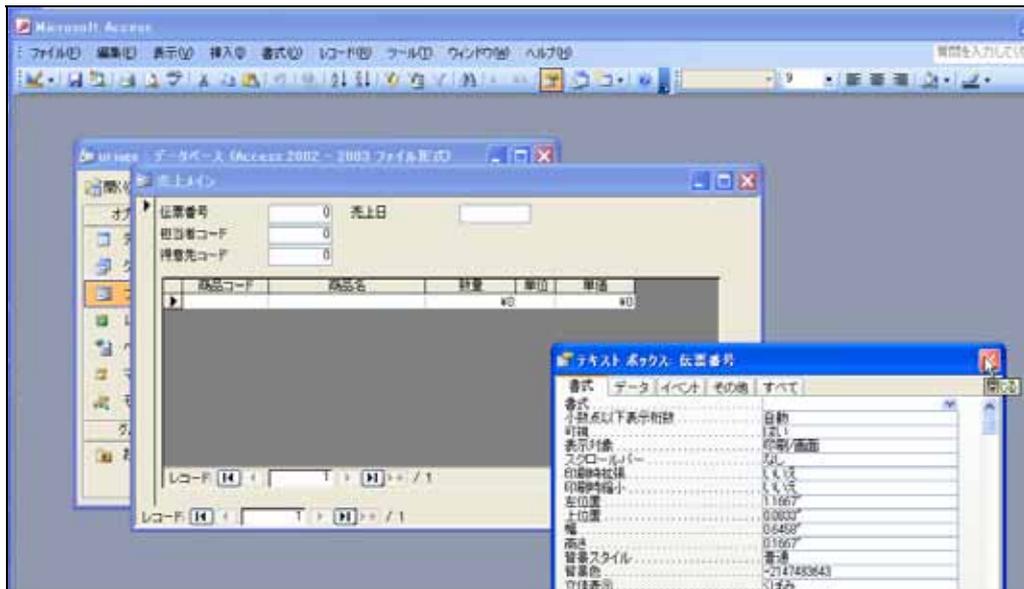
25. 再度、フォームをデザインビューに戻してレイアウト確認する。サブフォームのラベルが不適当な場所に表示されているので削除する。



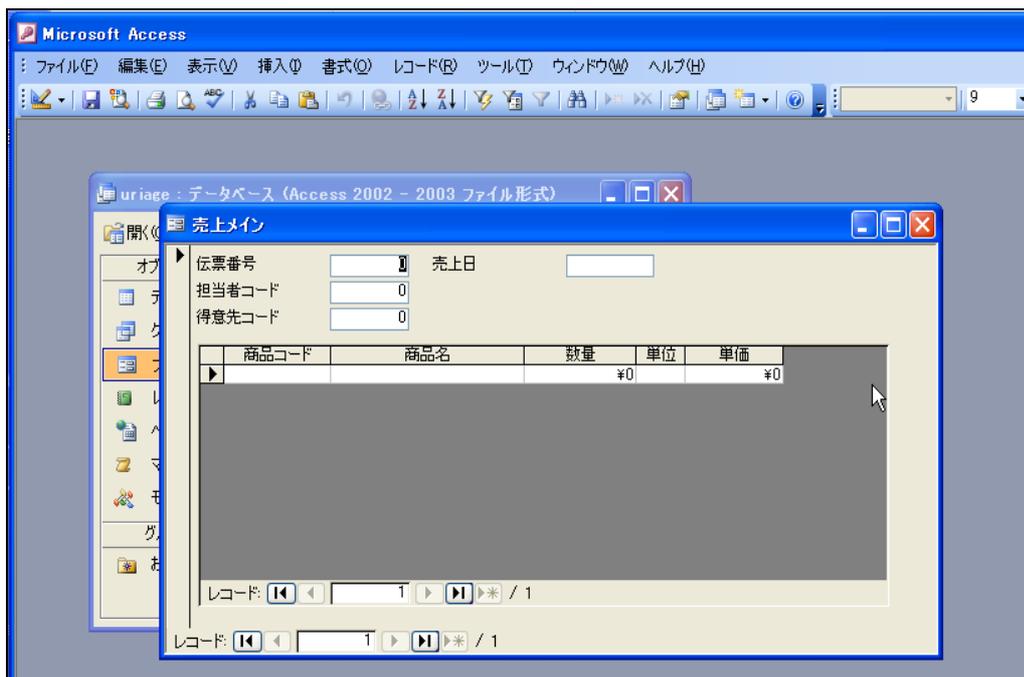
26. ここまで一度フォームを保存しておく。フォームの[閉じる]ボタンをクリックすると次のようなダイアログが表示される。[はい]ボタンをクリックする。



27. フォームが保存されたら、もう一度フォームを開いてみよう。場合によっては次のようにフォームのプロパティシートが表示されるかもしれない。



28. プロパティシートの[閉じる]ボタンをクリックするとシートは消える。この後は何度フォームを表示してもプロパティシートは表示されない。

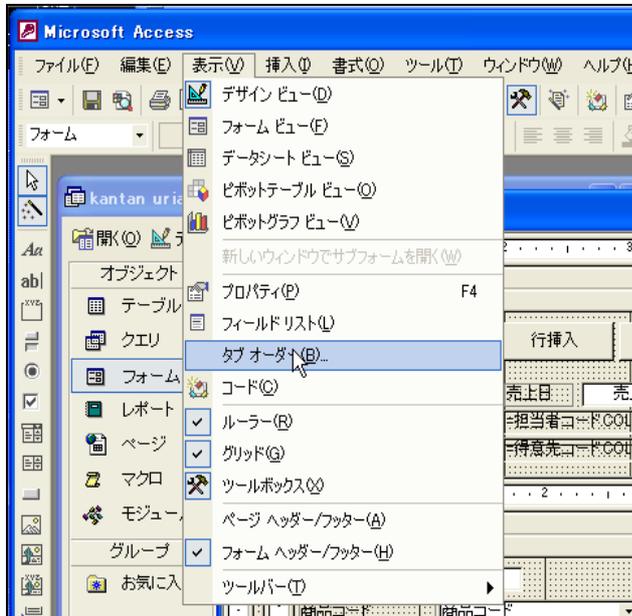


カーソルの移動順序を設定するには

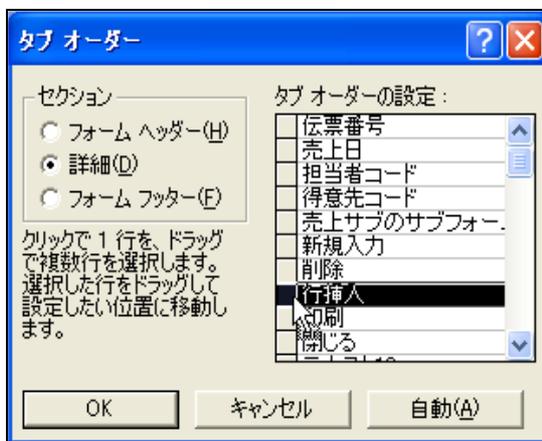
表示したフォームでカーソルの移動順序を確認してみよう。

思ったとおりに移動しない場合は、[タブオーダー]で変更する。手順は次の通り。

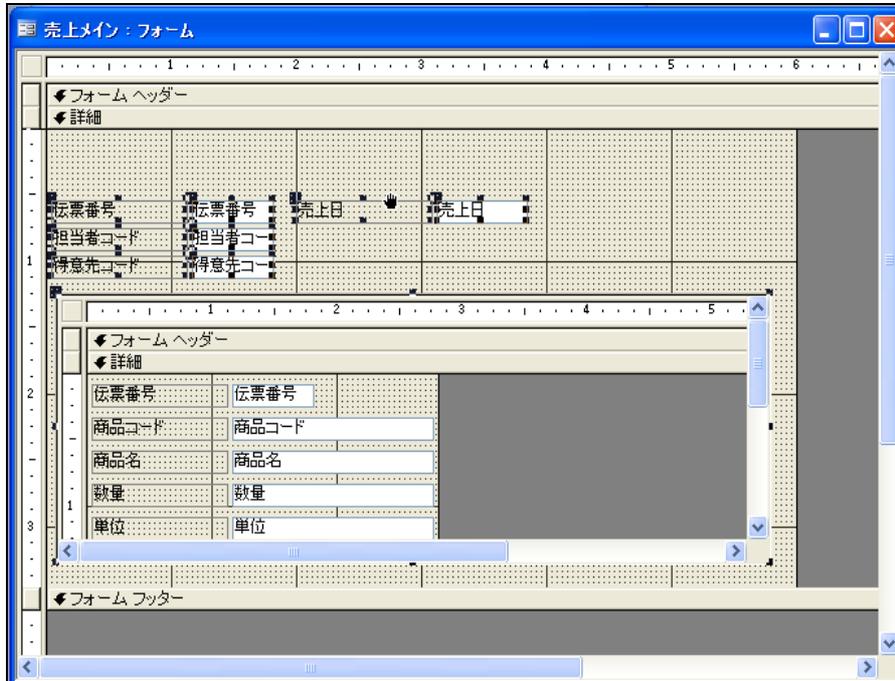
1. メニュー[表示]、[タブオーダー]をクリックする。



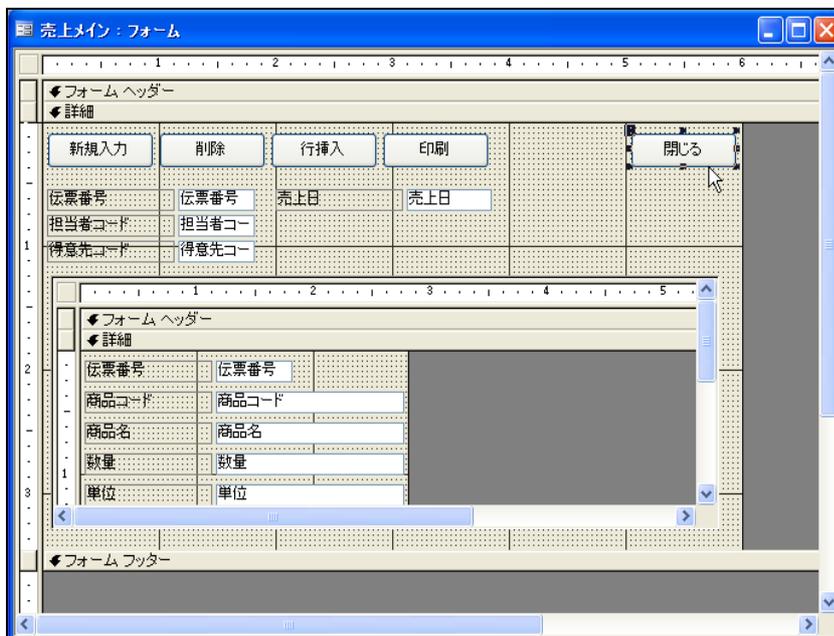
2. [タブオーダー]画面の[タブオーダーの設定]で移動順序を変更して[OK]ボタンをクリックする。



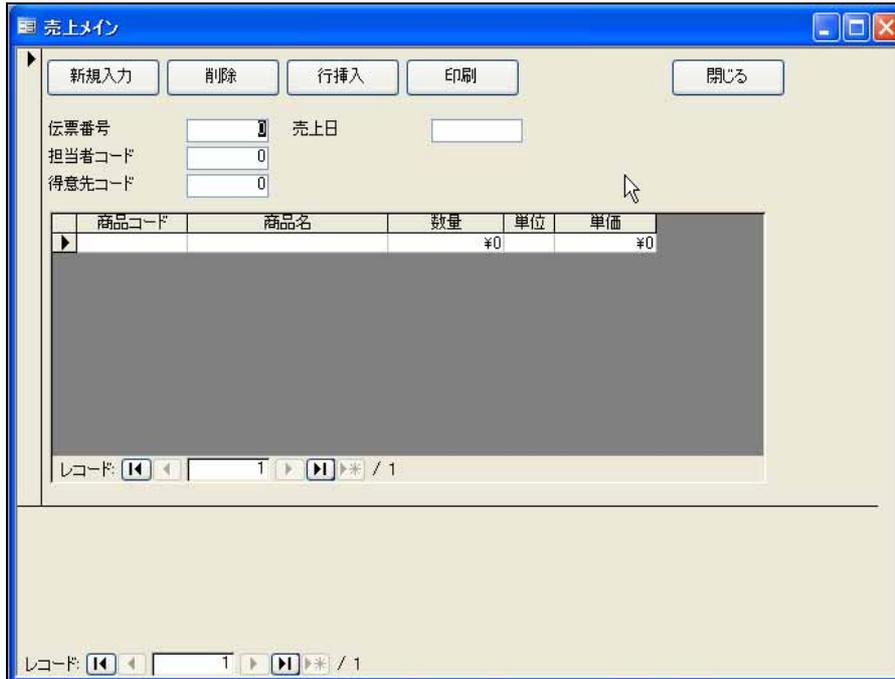
1. 次はフォームにコマンドボタンを配置して、サブフォームに[金額]フィールドを追加する。  
コマンドボタン配置用のスペースをフォーム上部に作成する。  
フォーム内にあるすべてのオブジェクトをマウスで範囲指定しドラッグすると一度に移動させることができる。



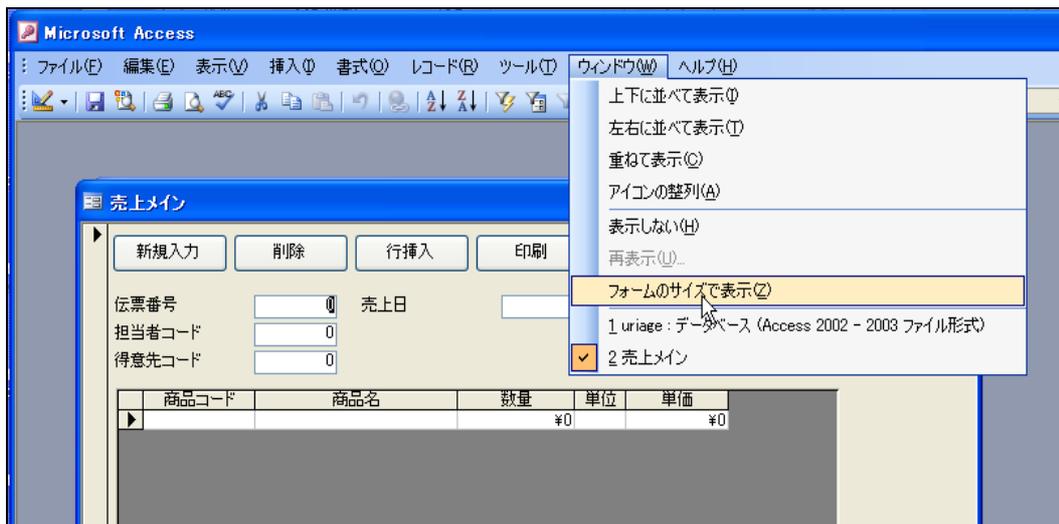
2. コマンドボタンを配置する。  
コマンドボタンは最初の1つを作り、サイズ等が決まったらそれをコピーして貼り付けを繰り返すとよい。



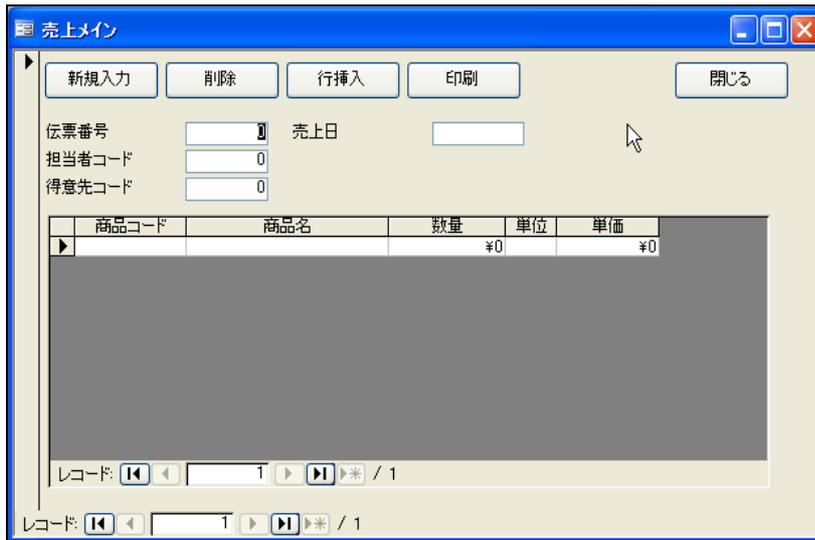
3. 一旦フォームを表示してみる。



4. フォームの下に余計なスペースが表示されたら、メニュー[ウィンドウ]、[フォームのサイズで表示]を実行して、もとのサイズを表示させる。



5. 元のサイズに表示された図。



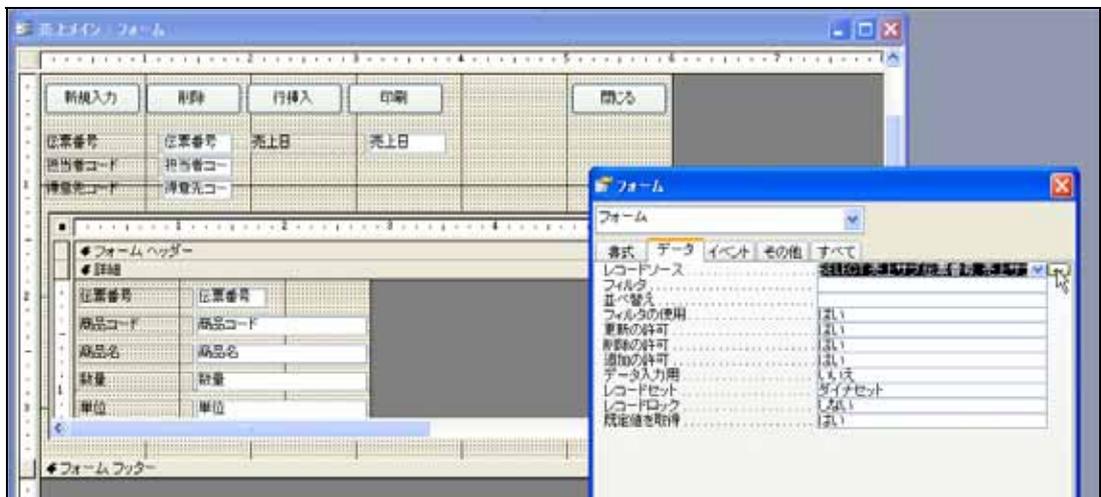
6. 続いて、サブフォームに[金額]フィールドを設定する。

サブフォームに[金額]を表示するためにはいくつかの方法があるが、ここでは基本的な方法を解説する。

まず、フォームのデザイン画面でサブフォームのプロパティを表示する（メインフォームのプロパティではない）。サブフォームのプロパティを表示するにはサブフォームの左上隅をダブルクリックするとよい。

プロパティの[データ]タブを選択し、[レコードソース]の右端アイコンをクリックしてクエリビルダを表示する。

式：



クエリビルダ画面



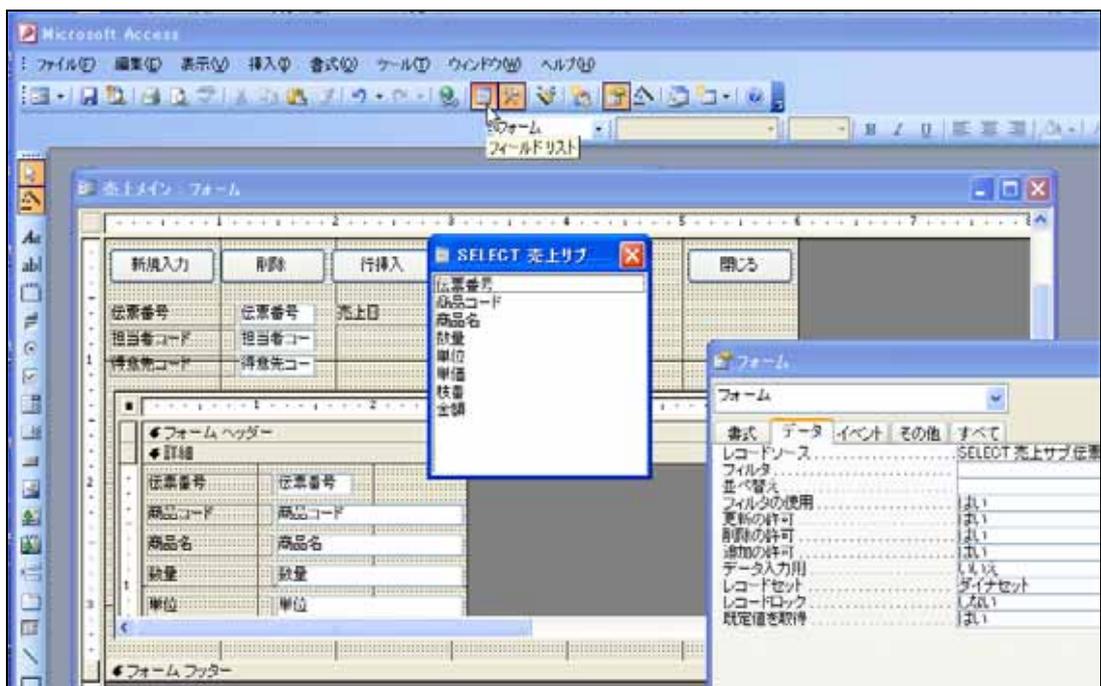
- クエリビルダのフィールド欄右端に下のような式を有した[金額]フィールドを追加する。式を登録したらビルダを閉じる。

式：金額: Fix(NZ([数量],0)\*NZ([単価],0))

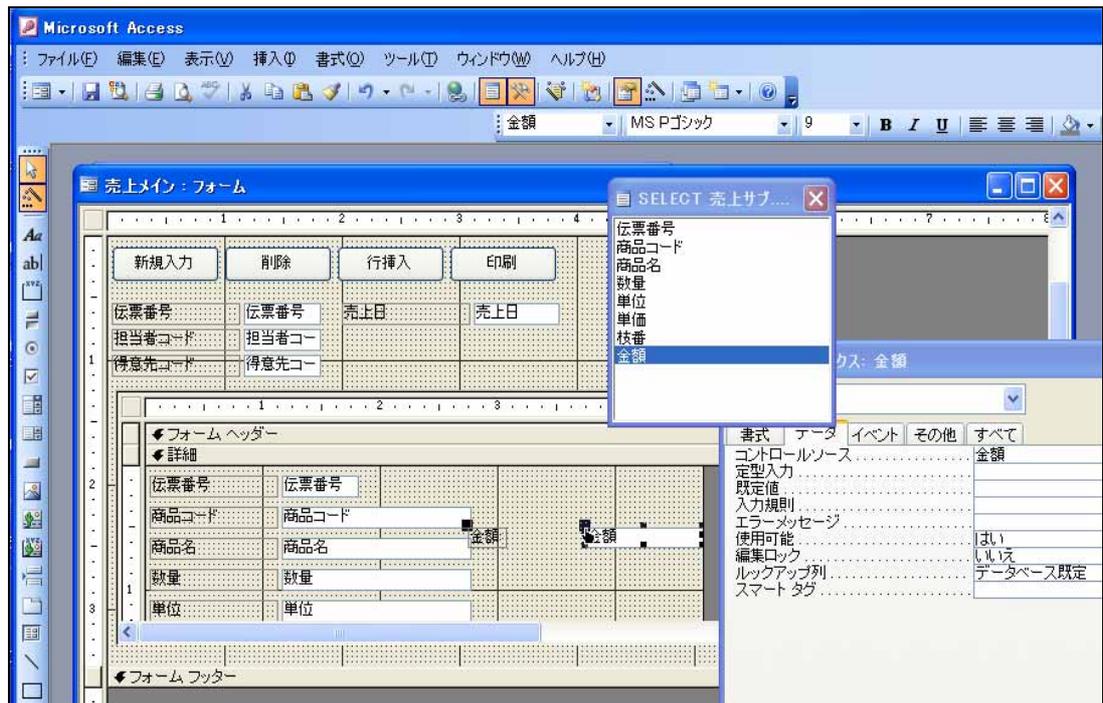
それぞれの関数については Access のヘルプを参照せよ



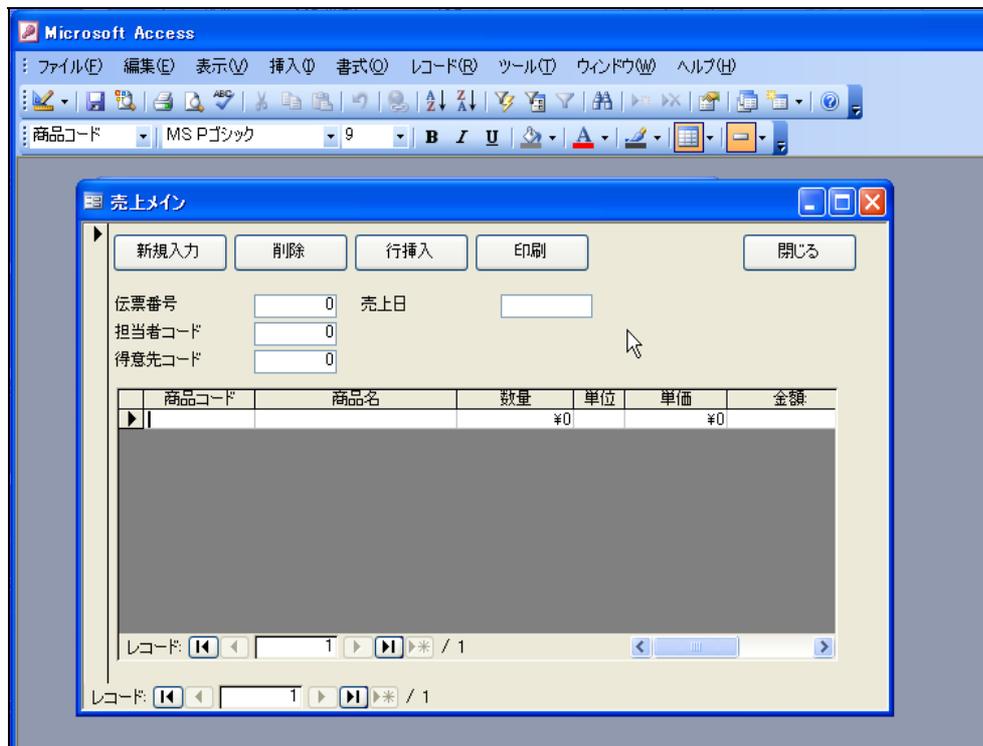
- フォームデザイン画面に戻ったら、[フィールドリスト]アイコンをクリックしてサブフォームのデータソースとなるクエリのフィールドを表示させる。



9. フィールドリストの中から[金額]フィールドをマウスでドラッグしてサブフォーム上に置く。  
置き場所はどこでもよい。



10. ここでフォームを表示してみる。[金額]フィールドが表示されていることを確認する。



## コンボボックスの作り方

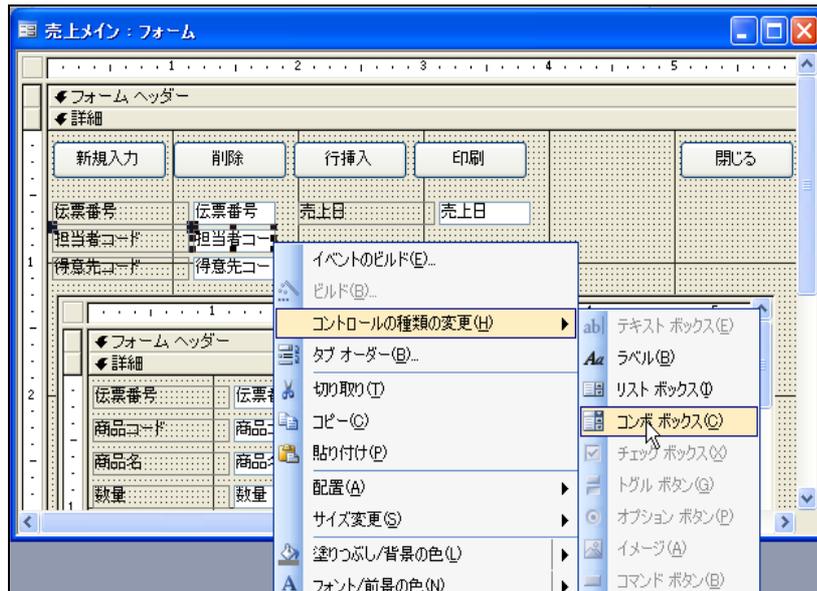
売上入力画面らしくなってきたが、まだまだ作業は続く。

次はコンボボックスの設定だ。

[担当者コード]、[得意先コード]、[商品コード]などはコンボボックスとして他のマスタから参照できるようにしたい。

[担当者コード]のコンボボックスの作り方は次の通り。

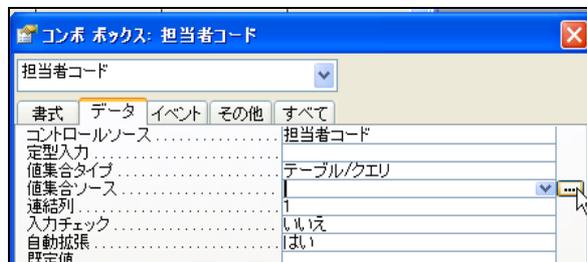
1. [担当者コード]テキストボックスを選択して、ショートカットメニュー(マウスの右ボタンクリック) [コントロールの種類変更]、[コンボボックス]を指定する。



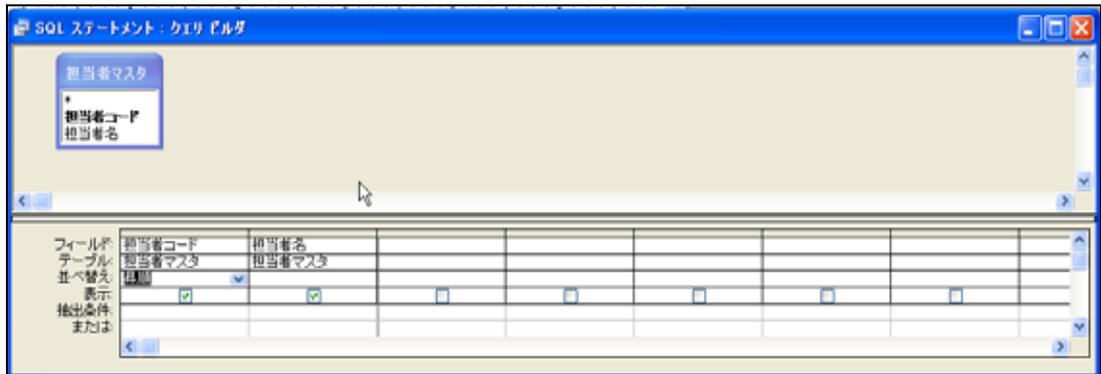
2. テキストボックスがコンボボックスに変更される。



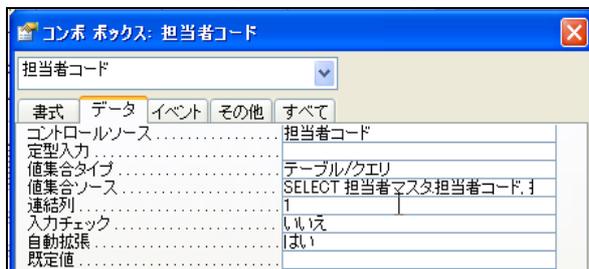
3. [担当者コード]テキストボックスのプロパティシートを表示し、[データ]タブをクリックする。次に[値集合ソース]右端のアイコンをクリックしてクエリビルダを表示する。



- クエリビルダで図のように設定する。[担当者コード]フィールドの[並べ替え]を『昇順』とする。



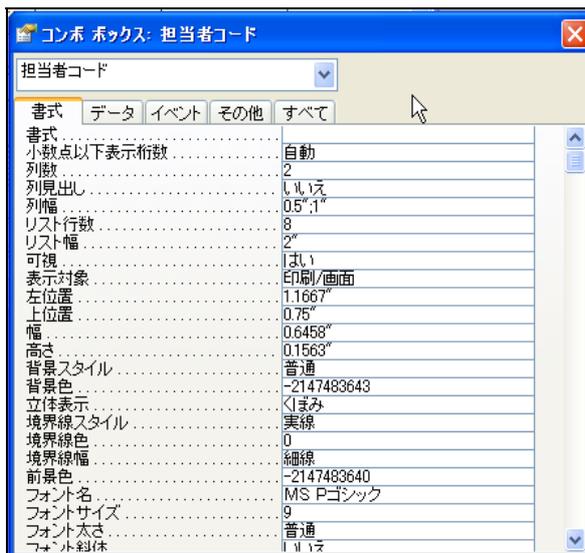
- クエリビルダを閉じて[値集合ソース]に文字列が登録されていることを確認する。



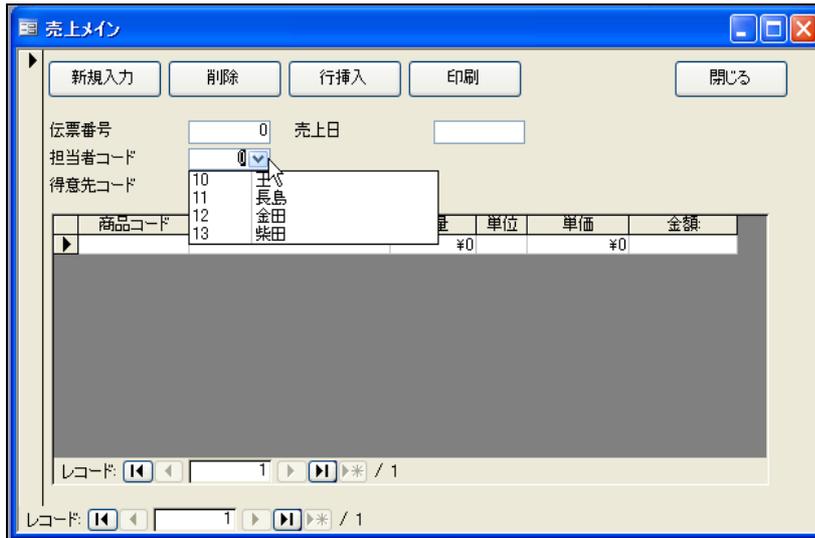
- プロパティシートの[書式]タブをクリックして、図のように設定する。

設定箇所	値
列数	2
列幅	0.5;1
リスト幅	2

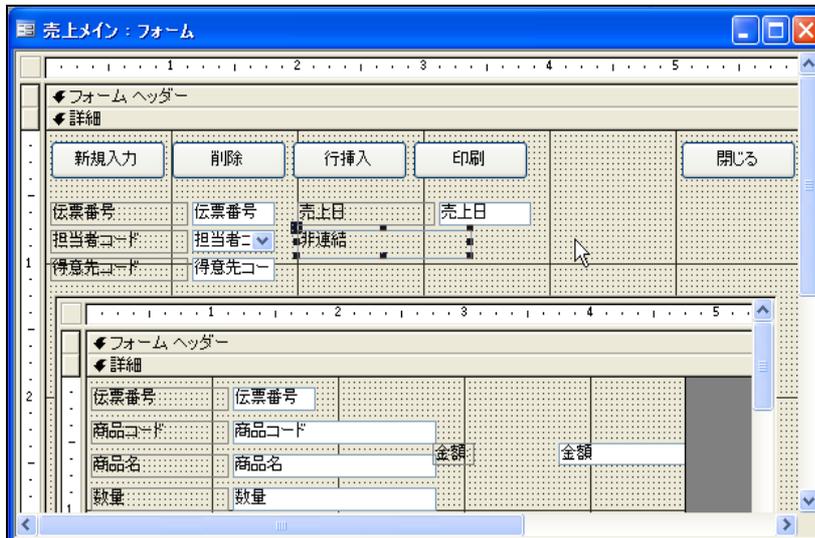
列幅では上記のように値をセミコロンで区切って数値のみを入力すると、自動的に『0.5";1"』と表示される。



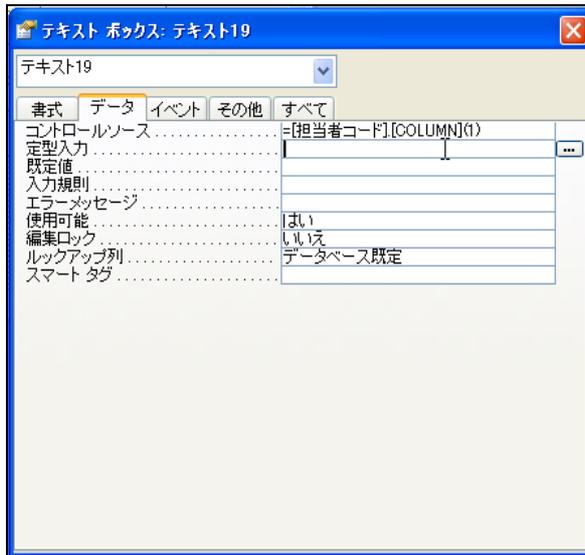
7. フォームを表示して動作を確認する。



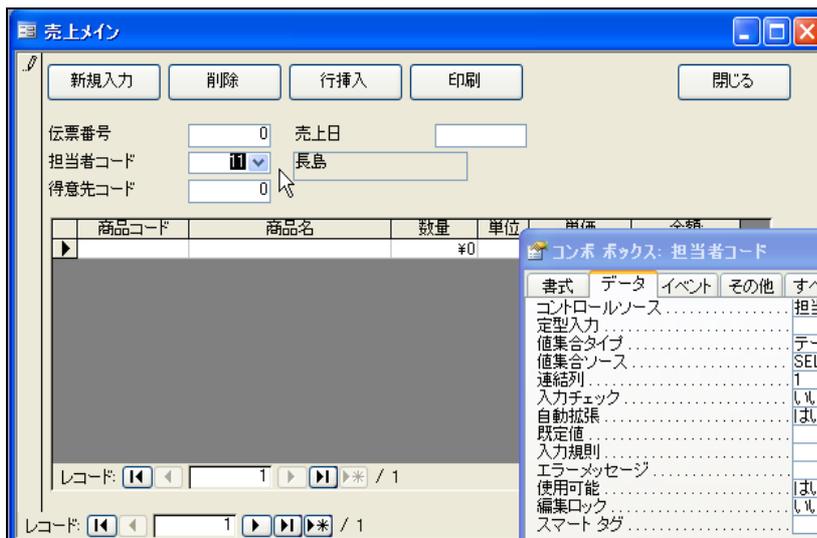
8. 正常に動作したら、次はコンボボックスの右横に名前を表示するテキストボックスを配置する。



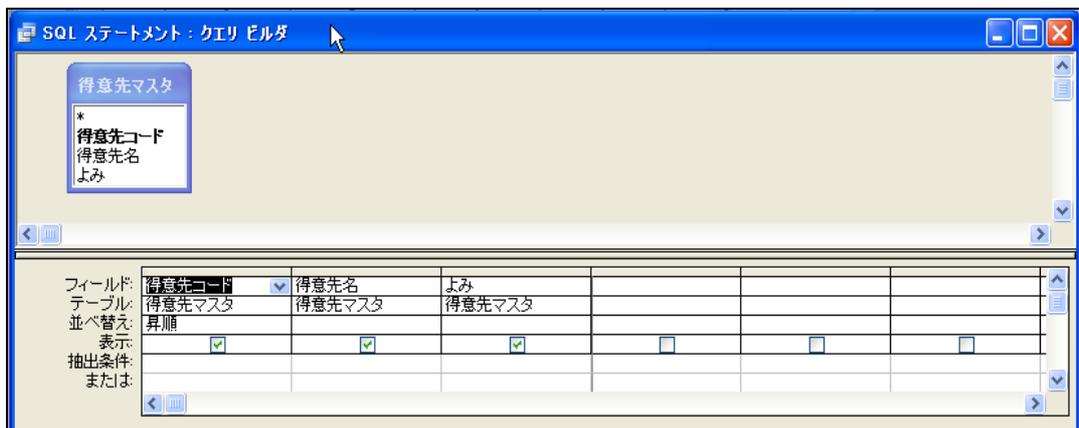
9. 追加したテキストボックスのプロパティで、[コントロールソース]を次のように設定する。  
 式：=担当者コード.COLUMN(1)  
 式の先頭に『=』を忘れずに記述する。



10. フォームを開いて動作を確認する。



11. 同様に、[得意先コード]、[商品コード]のコンボボックスも設定する。  
 [得意先コード]コンボボックスのクエリ

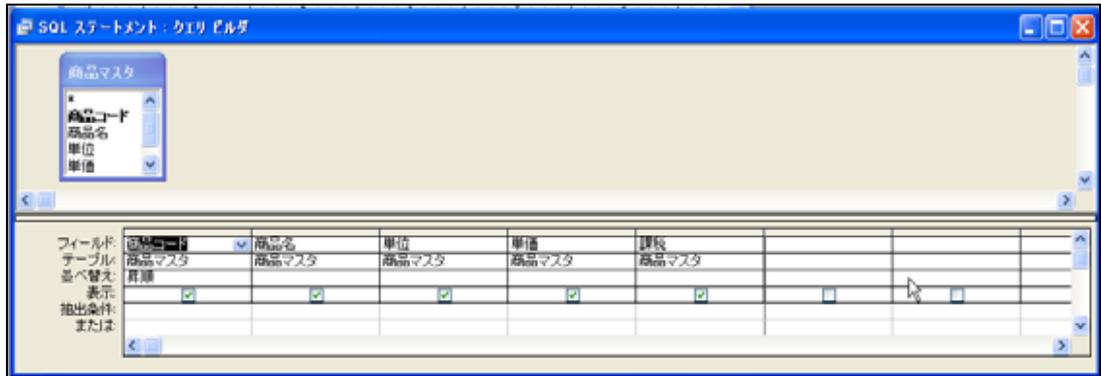


コンボボックスの並び順を「よみ」としたいときは、[よみ]に[昇順]の設定を行う。

[得意先コード]コンボボックスのプロパティ

設定箇所	値
列数	3
列幅	0.5;2;1
リスト幅	4

[商品コード]コンボボックスのクエリ

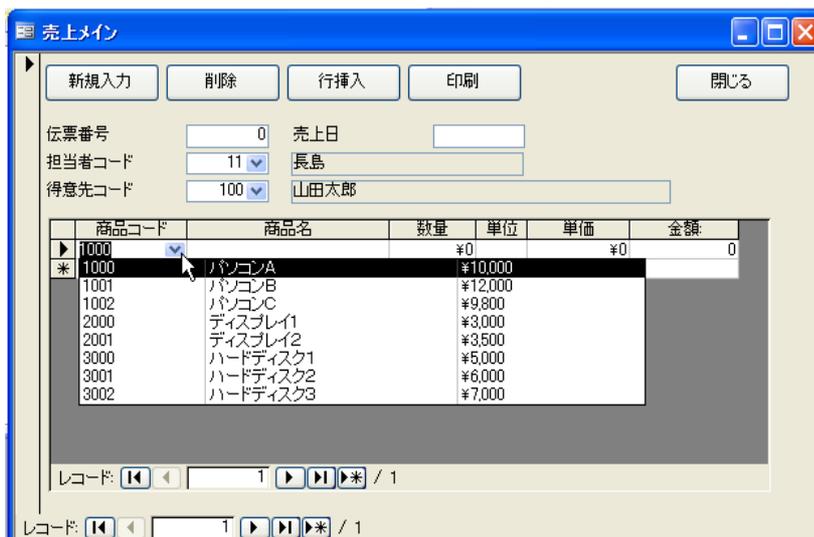


[商品コード]コンボボックスのプロパティ

設定箇所	値
列数	4
列幅	1;2;0;1
リスト幅	4.5

[単位]の列幅を0として非表示としている

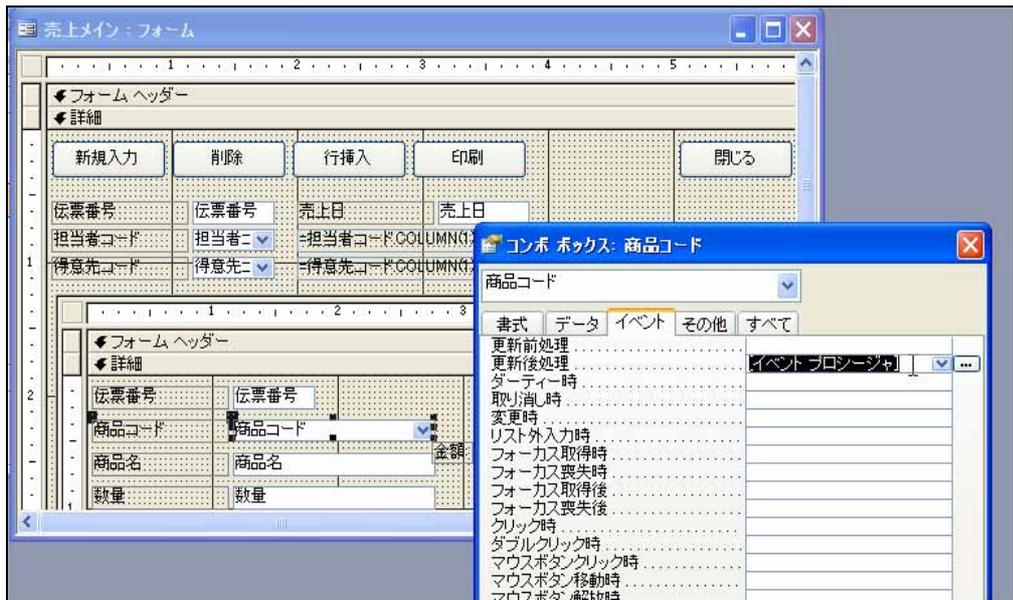
12. あらためてフォームを表示して動作を確認する。



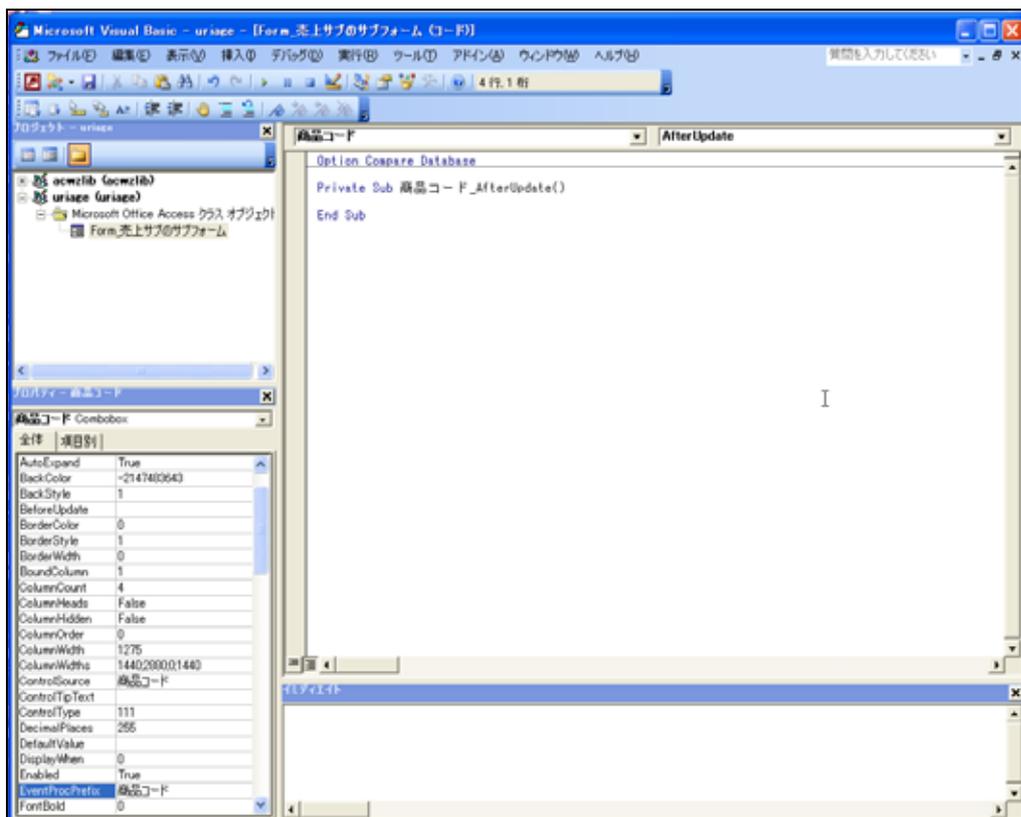
## イベントプロシージャの設定

サブフォームの[商品名]、[単位]、[単価]に[商品コード]コンボボックスのデータが複写されるよう、イベントプロシージャ（プログラム）を設定する。  
イベントプロシージャの設定手順は次の通り。

1. サブフォームの[商品コード]テキストボックスをダブルクリックしてプロパティシートを表示し、さらに[更新後処理]イベントの設定を[イベントプロシージャ]として右端のアイコンをクリックする。



2. イベントコード登録画面が表示される。



3. 『Private Sub 商品コード\_AfterUpdate()』以下に図に示すコードを記述する。

```

商品コード
AfterUpdate
Option Compare Database

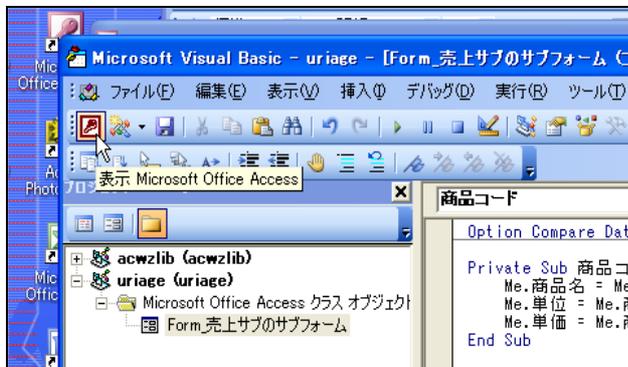
Private Sub 商品コード_AfterUpdate()
    Me.商品名 = Me.商品コード.Column(1)
    Me.単位 = Me.商品コード.Column(2)
    Me.単価 = Me.商品コード.Column(3)
End Sub
    
```

プロシージャは次の通り。

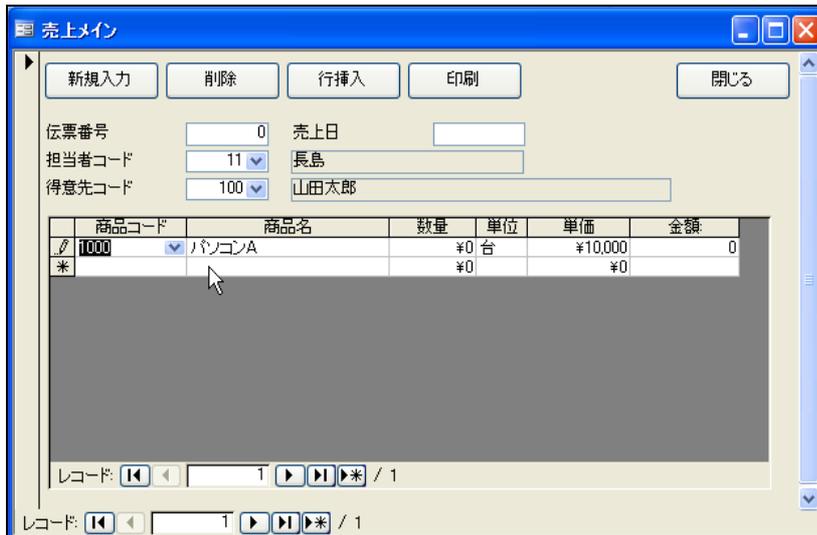
```

Private Sub 商品コード_AfterUpdate()
    Me.商品名 = Me.商品コード.Column(1)
    Me.単位 = Me.商品コード.Column(2)
    Me.単価 = Me.商品コード.Column(3)
End Sub
    
```

4. コードの記述が終了したら、ウィンドウ左上隅の[表示]アイコンをクリックしてフォームのデザイン画面に戻り、そこから再びフォームを表示してみる。



5. [商品コード]のコンボボックスを操作して、商品名、単位、単価が書き変わることを確認する。

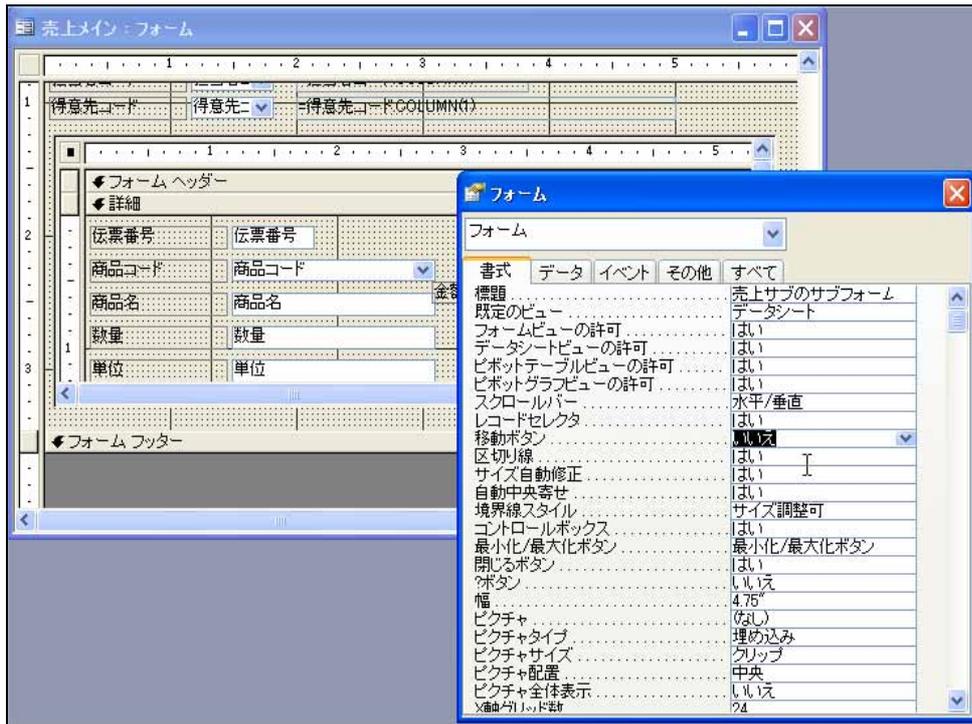


コンボボックスの設定が終了したら、その他の細かな設定を行う。

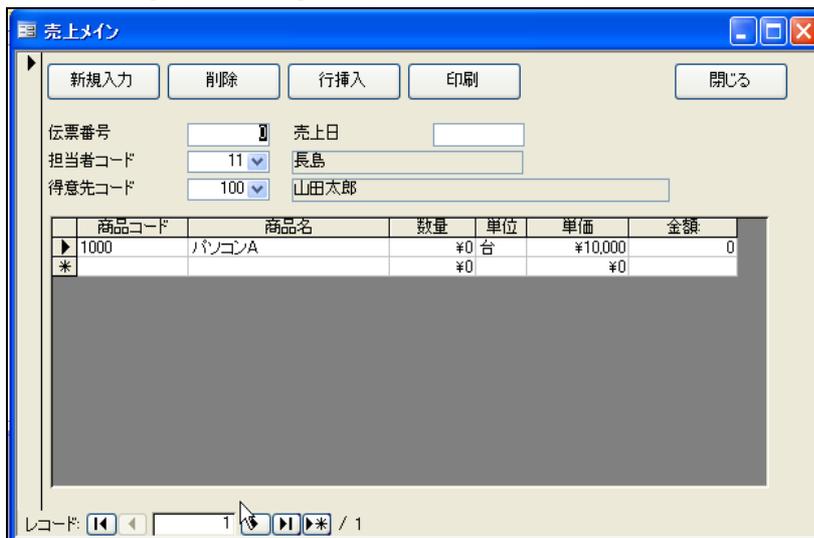
1. サブフォームの移動ボタンを消したい。



移動ボタンを消すには、サブフォームのプロパティシートを開いて、[書式]タブをクリックし、[移動ボタン]を『いいえ』に設定する。



2. サブフォームの[移動ボタン]が表示されなくなった。

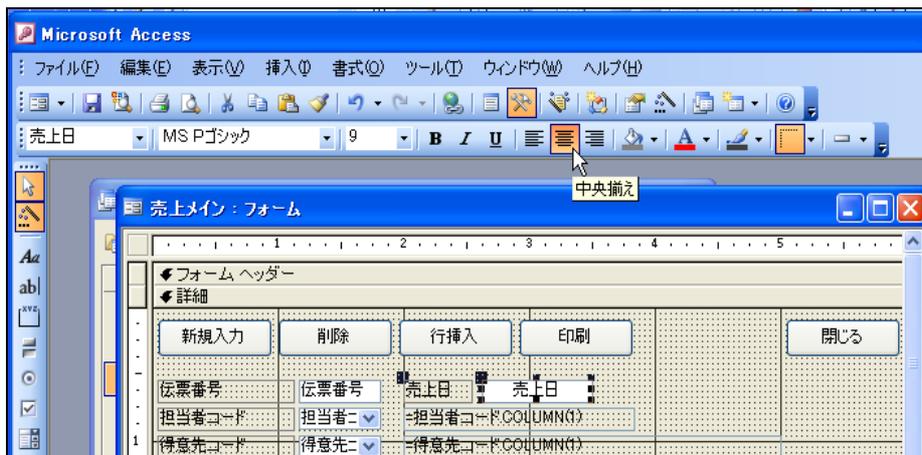


メインフォームの[移動ボタン]は残しておく。

- [売上日]テキストボックスの[定型入力]を設定する。  
式：9999/99/99;0;\_



- [売上日]を中央揃えに設定する。



- [売上日]にデータ入力して動作を確認する。



- フォームデザイン画面で、[数量]、[単価]、[金額]の[書式]を設定する。  
式：#,##0.#



7. フォームを開いて書式を確認する。

商品コード	商品名	数量	単位	単価	金額
1000	パソコンA	1.5	台	1,000,000.	1,500,000.

その他、気がついたところがあれば設定を行う。

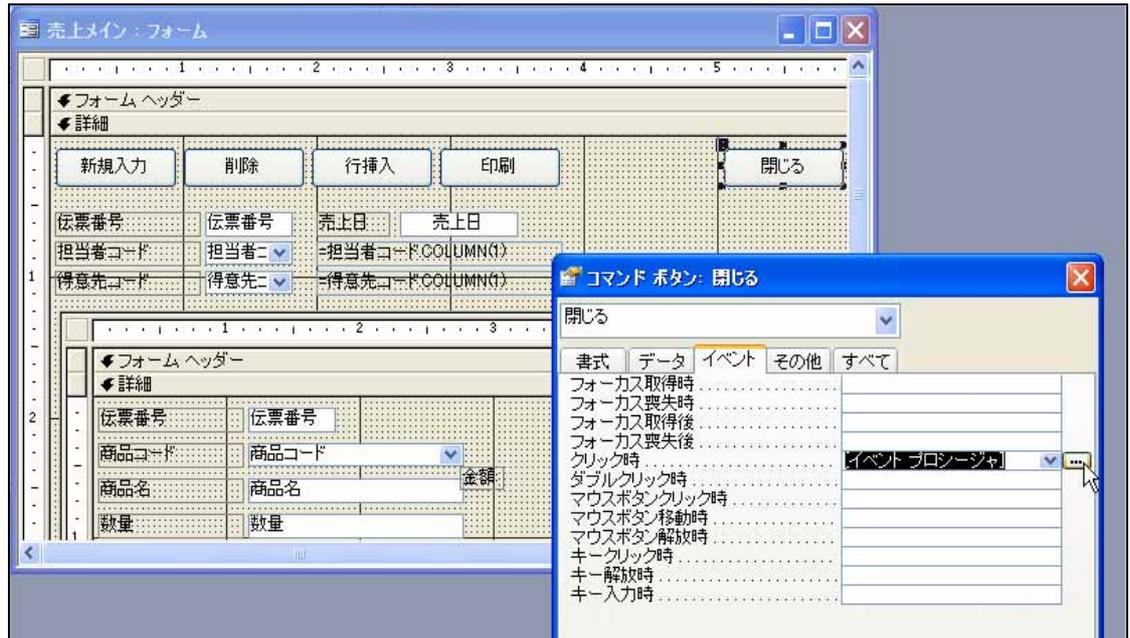
## コマンドボタンの設定

いよいよ本格的なプロシージャの設定に入る。

### [閉じる]ボタン

[閉じる]ボタンの機能はそのものずばり、フォームを閉じることだ。

フォームのデザインビューを開き、[閉じる]ボタンのプロパティシートを開き、[クリック時]のイベントプロシージャを記述する。



[閉じる]ボタンのイベントプロシージャは次の通り。

```
Private Sub 閉じる_Click()  
On Error GoTo exit1  
    'フォームを閉じる  
    DoCmd.Close acForm, Me.Name  
exit1:  
End Sub
```

### [新規入力]ボタン

[新規入力]ボタンが押されたら何を行うかをあらかじめ決めておく。  
ここでは次の動作を行う。

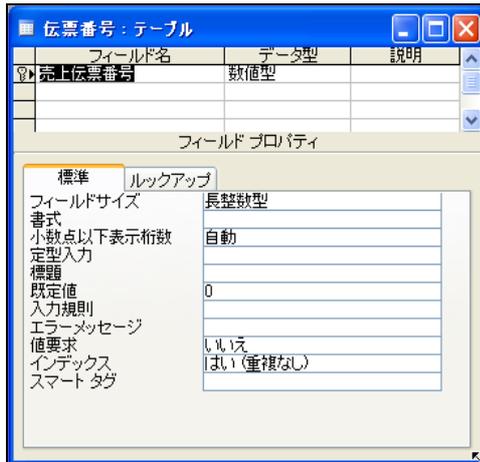
- ・ 新規の伝票番号を[伝票番号]テキストボックスに自動的に出力する。
- ・ [売上日]には自動的に本日の日付をセットする。
- ・ フォーカスを[売上日]に移動させる。

伝票番号は自動的にカウントアップされる必要がある。さらに、電源を切っても最後に利用した伝票番号はどこかで記憶している必要がある。

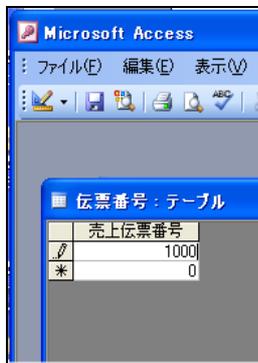
ということで、今回はデータベースシステムらしく伝票番号をテーブルに保存するという方法を

とることとした。

伝票番号を保存するための[伝票番号]テーブルを次のような設定として作成する。



テーブルを開いて、伝票番号の初期値としてとりあえず 1000 と入力して保存する。



フォームのデザインビューを開き、[新規入力]ボタンのプロパティシートを開き、[クリック時]のイベントプロシージャを記述する。

[新規入力]ボタンのイベントプロシージャは次の通り。

```
Private Sub 新規入力_Click()
'エラーが発生したら exit1 へ飛べ
On Error GoTo exit1
    '売上メイン]テーブルの新規レコードに移動する
    DoCmd.GoToRecord , , acNewRec
    '伝票番号を管理する[伝票番号]テーブルを開き、今ある値に+1 してその値をフォームの
    '[伝票番号]テキストボックスに複写する。
    Dim rst As DAO.Recordset
    Set rst = CurrentDb.OpenRecordset("伝票番号", dbOpenDynaset, dbSeeChanges)
    With rst
        .Edit
        !売上伝票番号 = !売上伝票番号 + 1
        .Update
        Me.伝票番号 = !売上伝票番号
        .Close
    End With
    Set rst = Nothing
    '売上日を本日付にセット
    Me.売上日 = Date
exit1:
End Sub
```

```
'フォーカスを売上日に移動
Me.売上日.SetFocus
Exit Sub
exit1:
'エラーが発生したらここに飛んでくる。
'通常はここにエラー処理を記述する。
End Sub
```

プロシージャの設定が終了したら動作確認を行う。

#### [削除]ボタン

[削除]ボタンがクリックされると次の動作が実行される。

- ・ 売上傳票の削除（メインフォームとサブフォームに表示されたデータの削除）

[削除]ボタンのイベントプロシージャは次の通り。

```
Private Sub 削除_Click()
'エラーが発生したら exit1 へ飛べ
On Error GoTo exit1
'新規レコードで削除を実行しようとしたらエラーメッセージを表示して作業を中止する
If Me.NewRecord And Me.伝票番号 = 0 Then
MsgBox "削除する伝票がありません。", , "削除エラー"
Me.削除.SetFocus
Else
'削除してもよいかを問うダイアログを表示
If MsgBox("現在表示中の伝票を削除してもよろしいですか?", vbYesNo + vbInformation +
vbDefaultButton2, "伝票削除") = vbYes Then
'不要なエラーメッセージが表示されないように
DoCmd.SetWarnings False
'次のステートメントで削除を実行
'この処理ではメインフォーム上の 1 レコードを削除するということになる
DoCmd.RunCommand acCmdDeleteRecord
DoCmd.SetWarnings True
End If
End If
Exit Sub
exit1:
MsgBox "削除エラー"
End Sub
```

フォームを開いて削除動作を確認してみよう。

[削除]ボタンがクリックされると確かに画面からはデータは消えた。しかし、テーブルを直接開いてみると[売上メイン]テーブルからはデータは削除されたが[売上サブ]テーブルのデータは残っている。なぜなら、この処理はあくまでもメインフォームのレコードソースとなるレコードを削除しているに過ぎないからだ。

このようなメインフォームとサブフォームが 1 対多の関係でリンクしたデータを削除するときは、通常メイン側のレコードを削除するのがセオリーである。処理としては正解だったのである。で

はなぜサブフォーム側が削除されなかったのか。

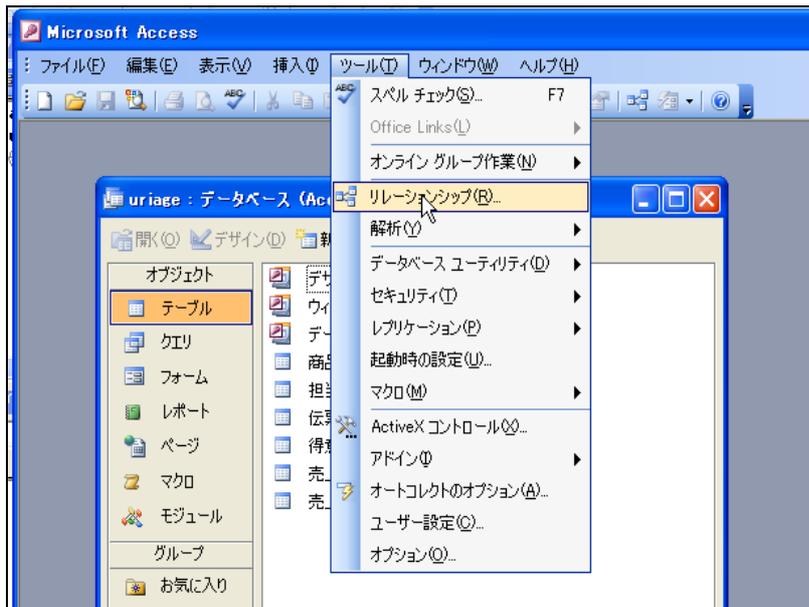
なぜなら、メインフォームとサブフォームとで、フォームとしてのリンクは行なわれていたがテーブル同士のリンクが行われていなかったからなのである。

[売上メイン]テーブルと[売上サブ]テーブルのリンクを設定するにはリレーションシップを設定しなければならない。

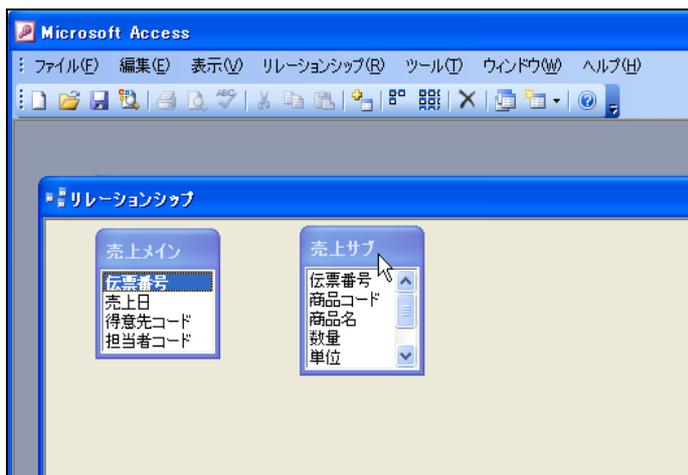
通常は、テーブルを定義した直後にこのリレーションシップを登録する。

リレーションシップの設定は次の通り。

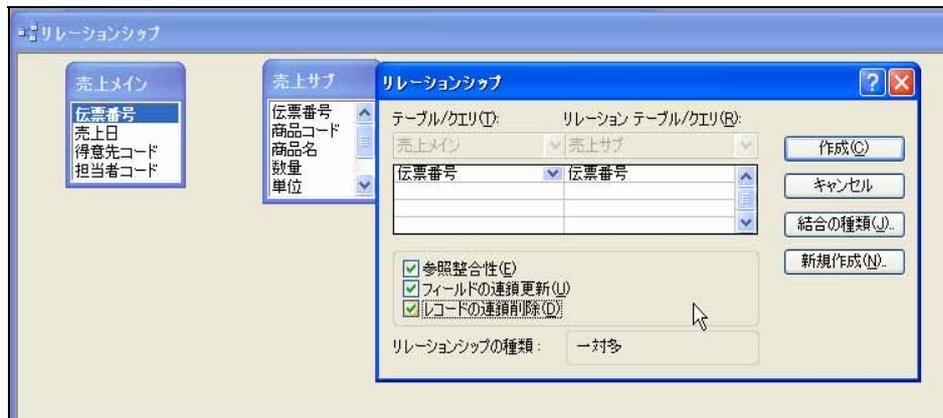
1. データベースウィンドウを表示した状態で、メニュー[ツール]、[リレーションシップ]を実行する。



2. リレーションシップウィンドウに[売上メイン]テーブルと[売上サブ]テーブルを表示させる。

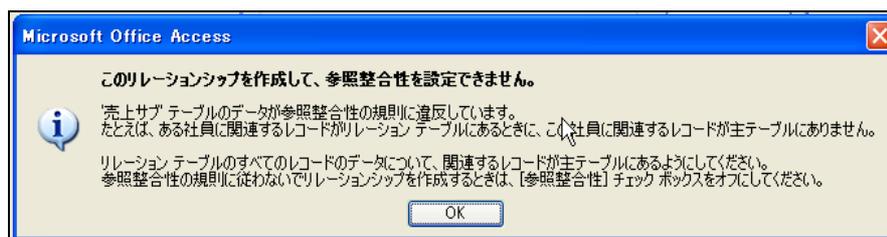


- [売上メイン]テーブルの[伝票番号]をマウスで選択し、そのまま[売上サブ]テーブルの[伝票番号]へとドラッグすると次のような画面が表示される。ここで、[参照整合性]と[フィールドの連鎖更新]、[レコードの連鎖削除]のチェックをすべてオンにする。最後に[作成]ボタンをクリックして作業を終了する。

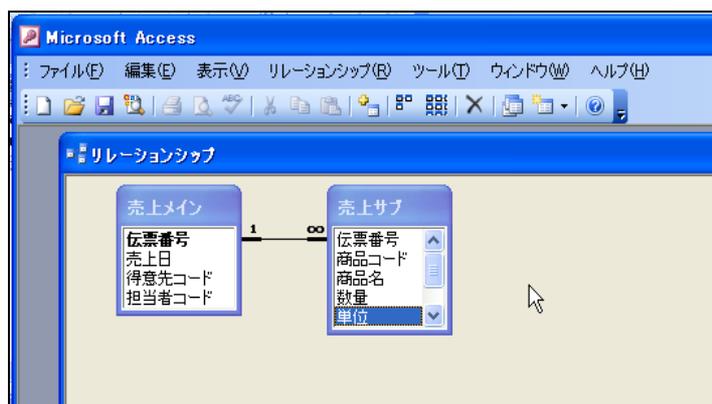


項目	解説
参照整合性	このチェックをオンにすると関連するテーブルの整合性が保障される。たとえば、一(一対多の側のこと)側テーブル側に存在しない[伝票番号]のレコードを多側に作成することはできなくなる。多側テーブルを直接開いて変更しようとしても変更後の値が一側に存在しないと拒否される。
フィールドの連鎖更新	ここをオンにすると、たとえば一側の[伝票番号]を変更すると、多側のテーブルの[伝票番号]が自動的にその値に置き換えられる。
レコードの連鎖削除	ここをオンにすると、一側を削除した場合に多側の関連レコードが自動的に削除される。

- [作成]ボタンがクリックされたときに次のような画面が表示される可能性があるが、これはすでに入力されているデータそのものに不整合があった場合だ。[売上メイン]テーブルと[売上サブ]テーブルを直接開いてすべてのデータを削除してから再度同様の操作を行えば問題ない。



- リレーションシップの登録が正常に終了すると次のような画面となる。



6. フォームを開いて[削除]ボタンをクリックし、該当するデータがすべて削除されているかを確認する。

### [印刷]ボタン

[印刷]ボタンは伝票を印刷するためのボタンだ。ボタンがクリックされると、『印刷しますか?』というダイアログを表示し、[OK]ボタンがクリックされると印刷が実行される。  
[印刷]ボタンのイベントプロシージャは次の通り。

```
Private Sub 印刷_Click()  
'印刷中に不用意なエラーメッセージを表示させないための制御  
On Error Resume Next  
    '表示中のデータを保存してから印刷を行う  
    DoCmd.RunCommand acCmdSaveRecord  
    If MsgBox("表示中の伝票を印刷しますか?", vbYesNo + vbInformation + vbDefaultButton2, "印刷") =  
vbYes Then  
        '次のステートメントで印刷が実行される  
        DoCmd.OpenReport "納品書", acViewPreview, "[売上メイン].[伝票番号]=forms![売上メイン].[伝票  
番号]"  
    End If  
End Sub
```

### [行挿入]ボタン

[行挿入]ボタンがクリックされると、サブフォーム上のカーソル位置に行が挿入される。  
[行挿入]ボタンのイベントプロシージャは次の通り。

```
Private Sub 行挿入_Click()  
'行挿入を行うには枝番作成プロシージャが動作していなければならない。  
'枝番作成プロシージャはこのファイルのトップに登録されている  
On Error GoTo 挿入失敗  
    'メインフォームの新規レコード行の入力途中では行われない  
    If Me.NewRecord And Me.伝票番号 <> 0 Then  
        Exit Sub  
    Else  
        '作業用のレコードセットと変数を定義  
        Dim rst As DAO.Recordset, 挿入番号 As Single  
        'サブフォームの明細データをレコードセットに代入  
        Set rst = Me.売上サブのサブフォーム.Form.RecordsetClone  
        '明細行の確認  
        If rst.RecordCount = 6 Then  
            Beep  
            MsgBox "明細行は6行までです!!", "確認"  
            'レコードセットを閉じる  
            rst.Close  
            'メモリーを開放する  
            Set rst = Nothing
```

```

        'プロシージャを抜ける
        Exit Sub
    End If
    '新規レコードかどうかの確認
    If Me.売上サブのサブフォーム.Form.NewRecord = True Then
        Beep
        MsgBox "新規レコードでは行挿入できません。"
        rst.Close
        Set rst = Nothing
        Exit Sub
    End If

    With rst
        'サブフォームの現在の位置をレコードセットの位置と同期させる
        .Bookmark = Me.売上サブのサブフォーム.Form.Bookmark
        'レコードセットに新規レコードを追加
        .AddNew
        ![伝票番号] = Me.伝票番号
        'レコードセットの[枝番]に、前行の枝番に 0.01 を加算した値を代入
        ![枝番] = .AbsolutePosition + 0.01
        'レコードセットを更新
        .Update
        '並べ替えを実行するためにリクエリする
        'これにより画面上では並べ変りが行われ、
        '新規レコードがいかにも挿入されたように見える
        Me.売上サブのサブフォーム.Requery
        '枝番を再度振りなおす
        Call 枝番作成
        'フォーカスをサブフォームにセット
        Me.売上サブのサブフォーム.SetFocus
        'さらにフォーカスを商品コードにセット
        Me.売上サブのサブフォーム.Form![商品コード].SetFocus
        'レコードセットを閉じる
        .Close
    End With
    'メモリを開放
    Set rst = Nothing
End If
'プロシージャを抜ける
Exit Sub
挿入失敗:
    MsgBox "挿入処理は無効です。"
End Sub

```

行挿入の原理は次の通り。

1：行挿入を行う前に次のような明細が存在していて、現在のカーソルは2行目にあるとする。

売上明細行	枝番
明細 1	1
明細 2	2
明細 3	3

2：実は、Access では（多くのデータベースでも同様）行挿入という概念はなく、行追加という手段しか存在しない。行追加が実行されると次のような状態となる。

売上明細行	枝番
明細 1	1
明細 2	2
明細 3	3
空白行	1.01

3：リクエリでテーブルの並べ替えを実行すると、見かけ上の行挿入が完了する。

売上明細行	枝番
明細 1	1
空白行	1.01
明細 2	2
明細 3	3

4：再度枝番を振りなおす。

売上明細行	枝番
明細 1	1
空白行	2
明細 2	3
明細 3	4

行挿入プロシージャの途中で明細行をカウントして、指定した行数に達した場合に処理を中止する設定となっている。

このレコード数のチェックは、サブフォームの[挿入前処理]イベントでも次のように設定しておかねばならない。

```
Private Sub Form_BeforeInsert(Cancel As Integer)
'この後のプログラムでエラーが発生したら exit1 へ飛べ
On Error GoTo exit1
    '作業用レコードセットを宣言
    Dim rst As DAO.Recordset
    '現在のサブフォームの状態を rst に代入
    Set rst = Me.RecordsetClone
    With rst
        If .RecordCount = 6 Then
```

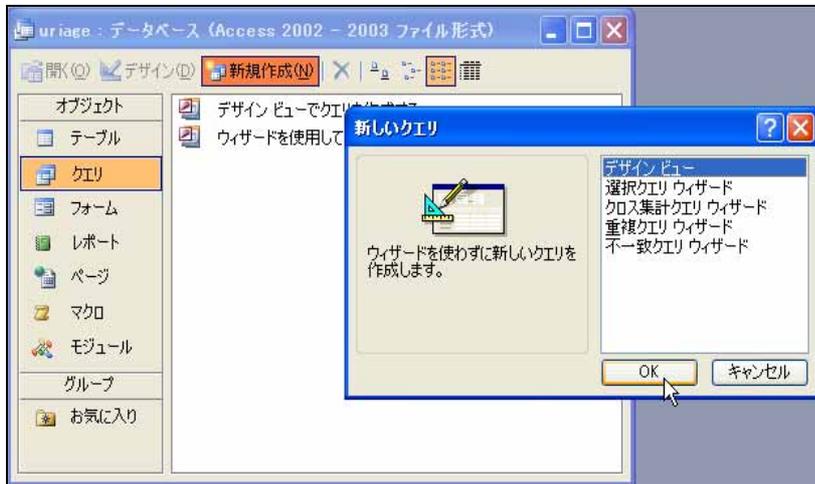
```
        Beep
        MsgBox "明細行は6行までです!!", "確認"
        'cancel に True を代入すると行挿入が無効となる
        Cancel = True
    End If
    'レコードセットを閉じる
    .Close
End With
'メモリーを解放
Set rst = Nothing
exit 1:
End Sub
```

## 納品書レポートの作成

ここではもっともシンプルなレポート作成方法をご紹介します。

はじめにレポート作成で必要となるクエリを作成する。クエリ作成手順は次の通り。

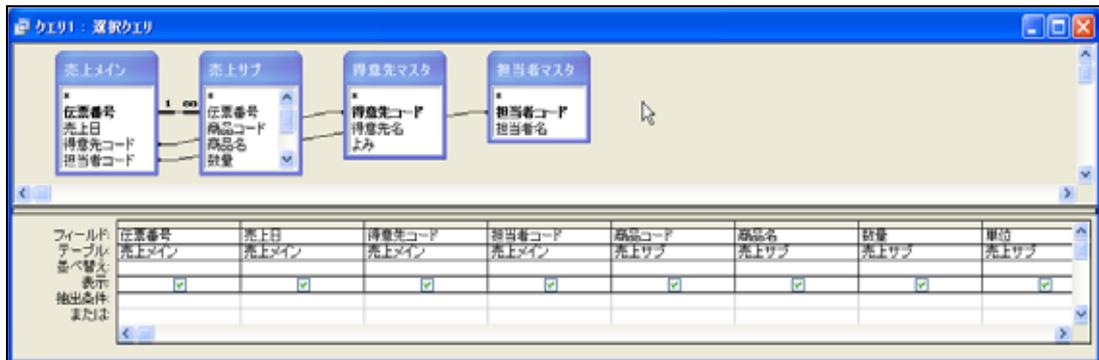
1. データベースウィンドウの[クエリ]を選択し、[新規作成]をクリックする。  
[新しいクエリ]画面で[デザインビュー]を選択して[OK]ボタンをクリックする。



2. [テーブルの表示]画面で[売上メイン]、[売上サブ]、[得意先マスタ]、[担当者マスタ]の各テーブルを追加する。



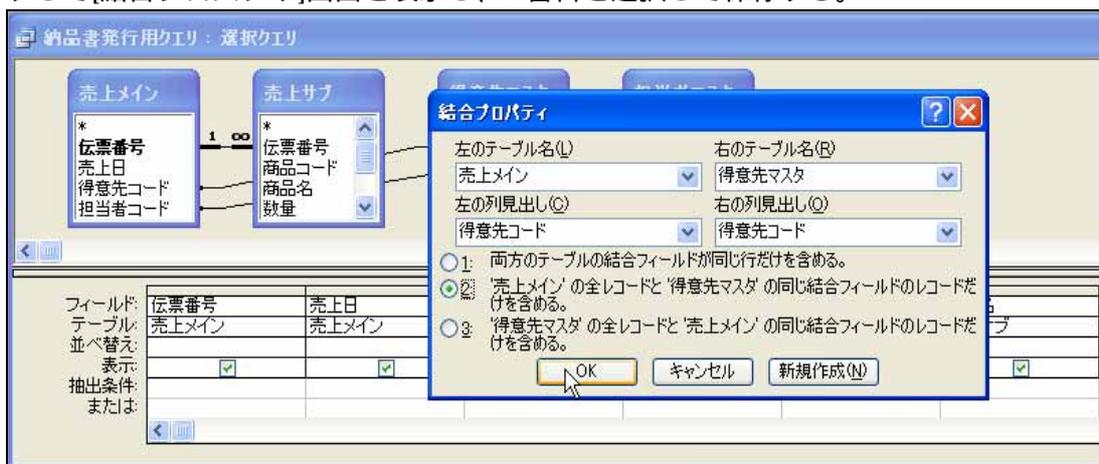
- [売上メイン]、[売上サブ]テーブルからすべての項目をフィールド欄に登録する(ただし、[売上サブ]テーブルの[伝票番号]だけは除外する)。続けて、[得意先マスタ]テーブルの[得意先名]、[担当者マスタ]テーブルの[担当者名]を追加する。



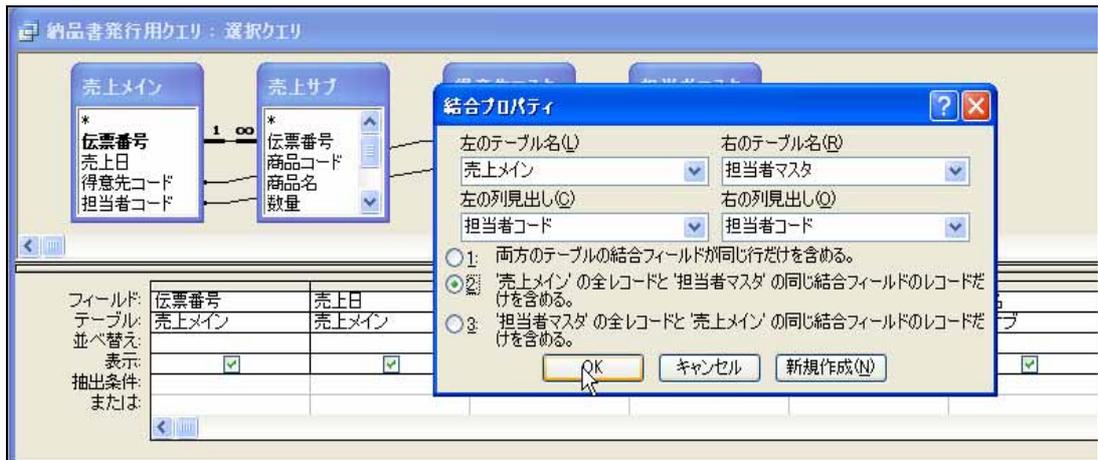
- フィールド欄に金額を表示するためのフィールドを追加する。この作業は[売上サブのサブフォーム]フォームで作成した[金額]フィールドと同じである。式をそのまま複写してもよい。



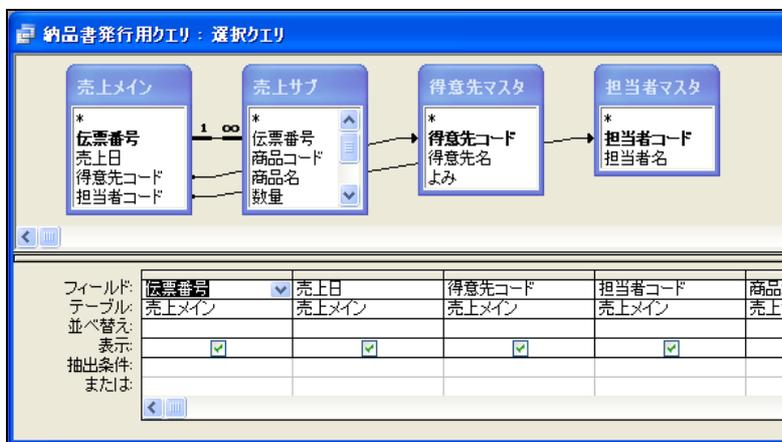
- [売上メイン]テーブルの[得意先コード]から出ているリレーションを示すラインをダブルクリックして[結合プロパティ]画面を表示し、2番目を選択して保存する。



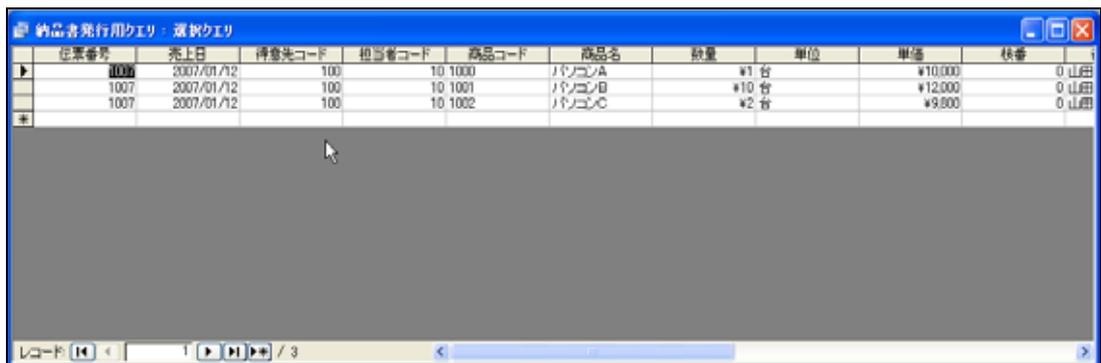
- 同じ要領で、[担当者コード]から出ているラインをダブルクリックして[結合プロパティ]画面を表示し、2番目を選択して保存する。



- 操作が正常に行われると次のような画面となる。



- 出来上がったクエリを表示してみてデータが正常に表示されるか確認する。



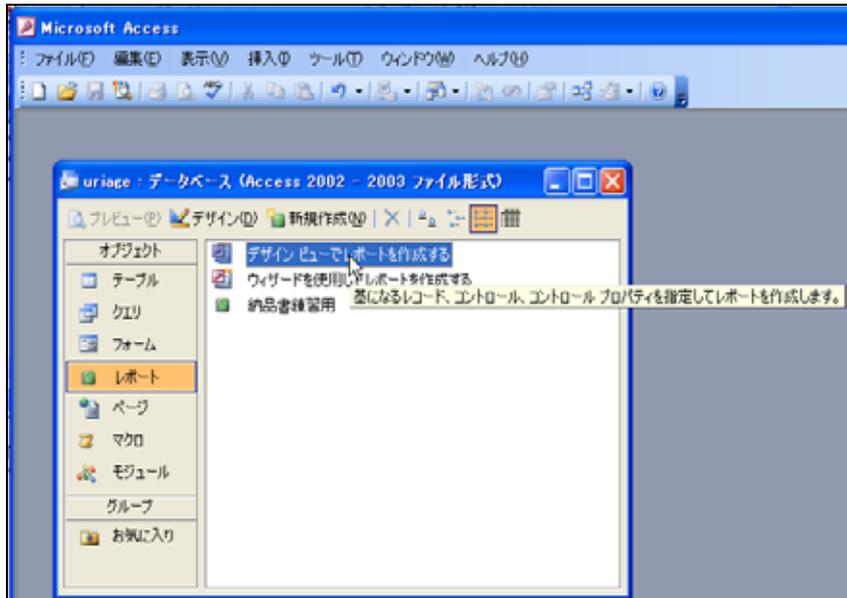
- クエリに名前を付けて保存する。

次はいよいよレポートの作成だ。

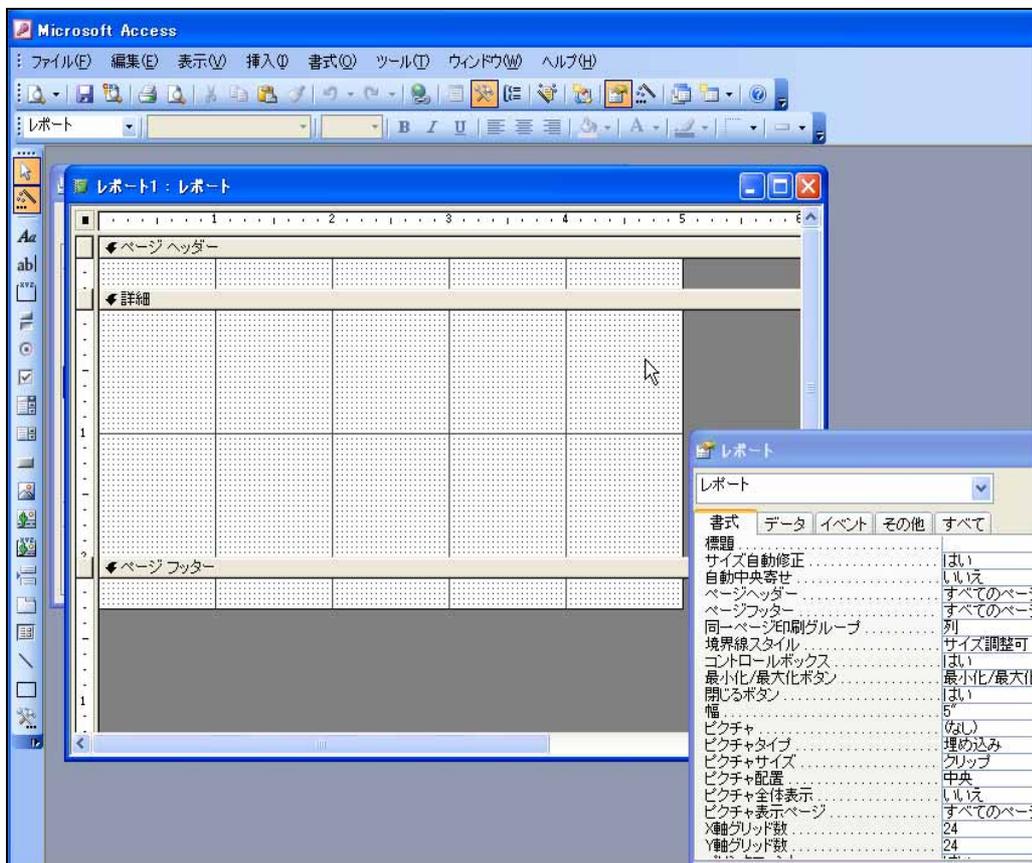
Access のレポート作成にはいろいろな方法があり、今回ご紹介する方法はその中の一つの方法である。初心者のためのレポート作成ウィザードも用意されているが、あれなどは初心者をよけいに惑わせる要素があると思われるのでお勧めはしない。

レポートの作成方法は次の通り。

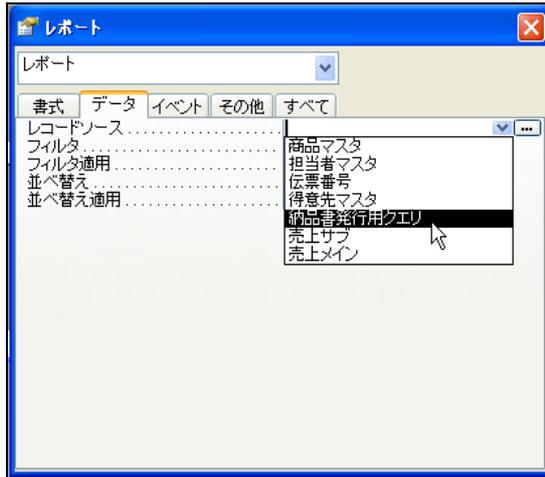
10. データベースウィンドウの[レポート]を選択し、[デザインビューでレポートを作成する]をクリックする。



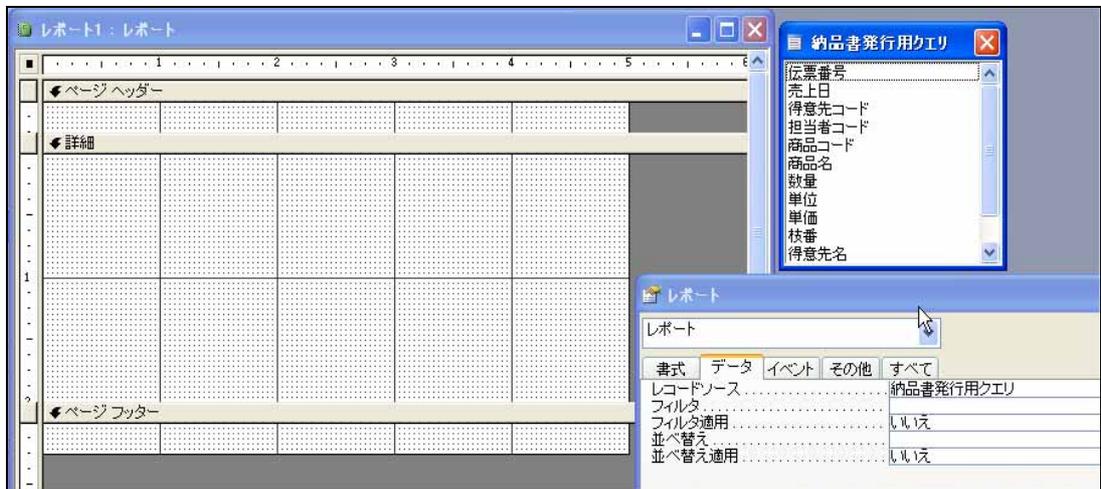
11. レポートデザイン画面が表示される。



12. レポートのプロパティシートを表示し、[レコードソース]で先ほど作成したクエリを指定する。



13. 図のようにクエリのフィールド一覧がウィンドウに表示される。



14. 最初にグループや並べ替え設定を行う。

通常、データを印刷する場合にはどういったグループで印刷するか、どのような順序で印刷するかを指定しなければならない。画面上部の[並べ替え/グループ化の設定]アイコンをクリックする。

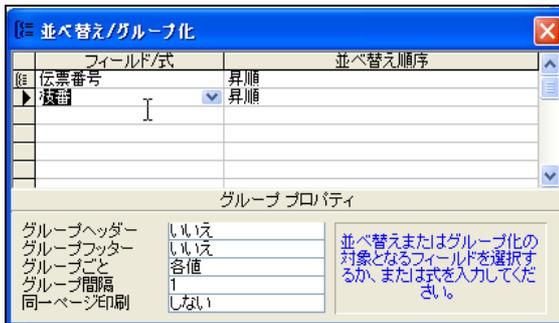


15. 印刷順序をここで指定する。

最初に[伝票番号]を登録し、[グループヘッダー]、[グループフッター]の表示を『はい』とする。



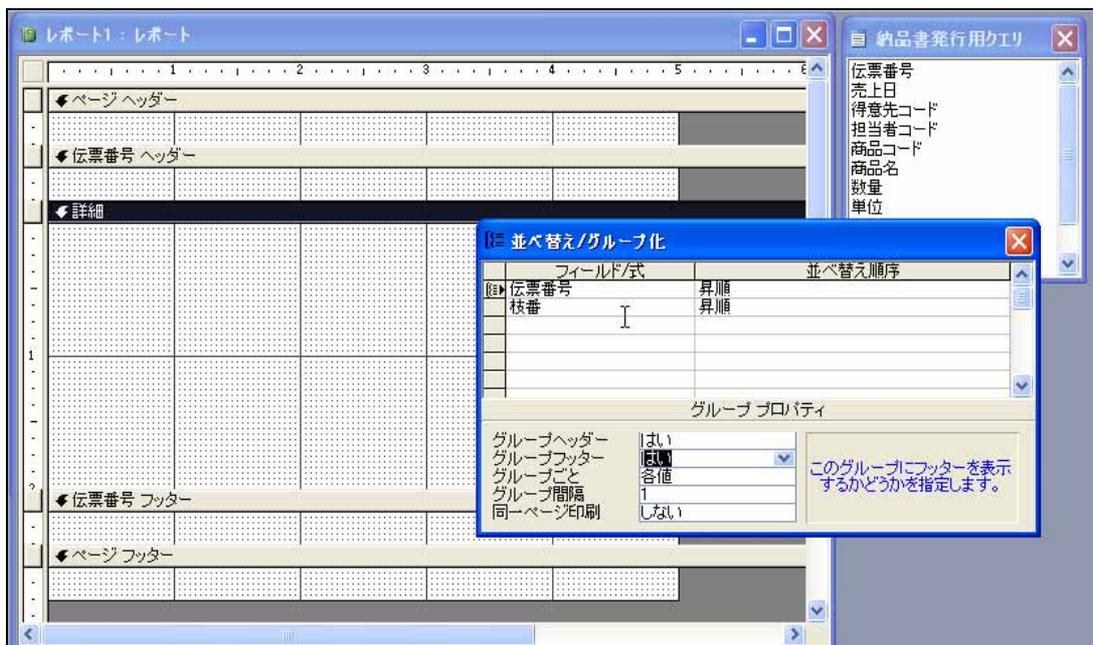
16. 同じように[枝番]を 2 番目に登録してこの画面を閉じる。



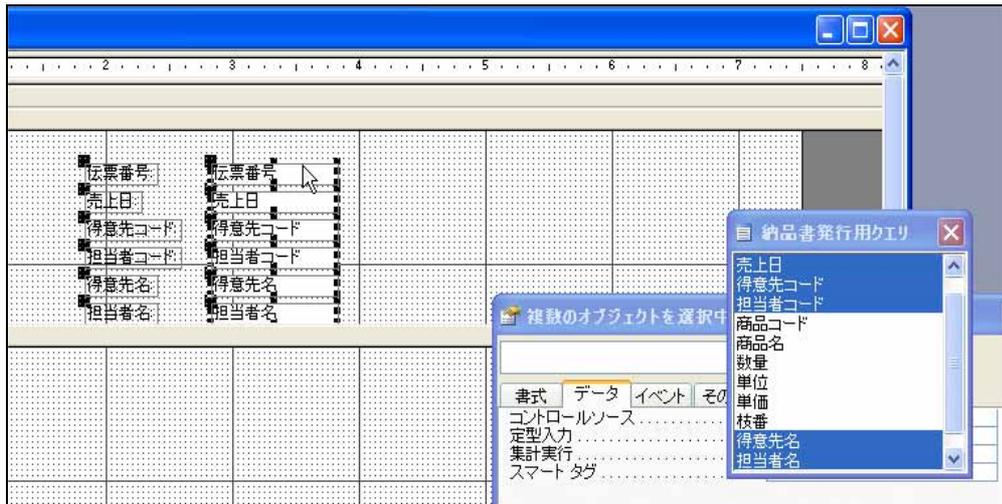
伝票番号ごとの枝番順に印刷されるということになる。

17. レポート画面には[伝票番号ヘッダー]部と[伝票番号フッター]部が追加されていることがわかる。

[ページヘッダー]、[ページフッター]は今回必要ないので高さを0としておく。



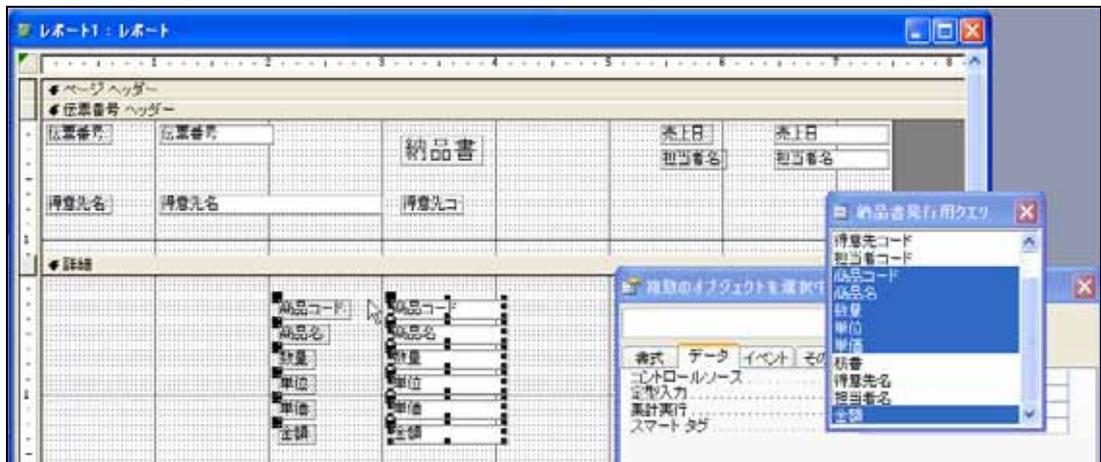
18. フィールド一覧ウィンドウから[伝票番号ヘッダー]部に必要なフィールドを登録する。マウスのドラッグ&ドロップ操作を使うと効率的だ。



19. レイアウトを整える。



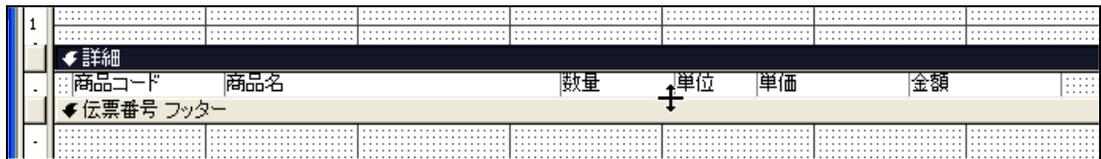
20. 同様に[詳細]部に必要なフィールドを登録する。



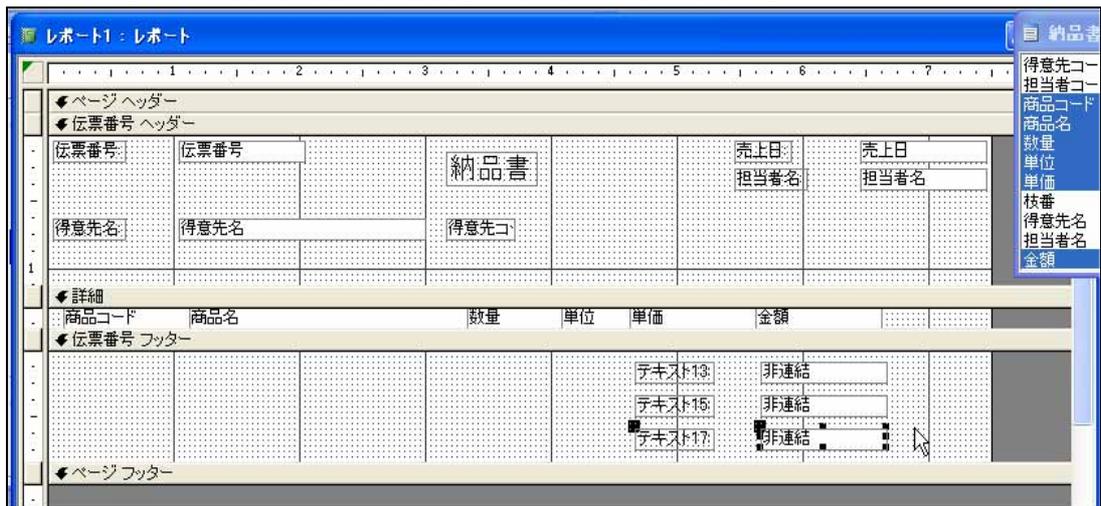
21. [詳細]部のレイアウトを整える。



22. [詳細]部の高さを調整する。ここの高さが伝票の1明細の高さとなる。



23. [伝票番号フッター]部にテキストボックスを3つ登録する。  
これらは、商品計、消費税、総合計を表示するためのものである。



24. ラベルを変更する。



25. 登録したテキストボックスに式を登録する。

商品計を表示するためのテキストボックスに次の式を登録する。

式: =Sum([金額])

テキストボックスの名前を『A』に変更する。

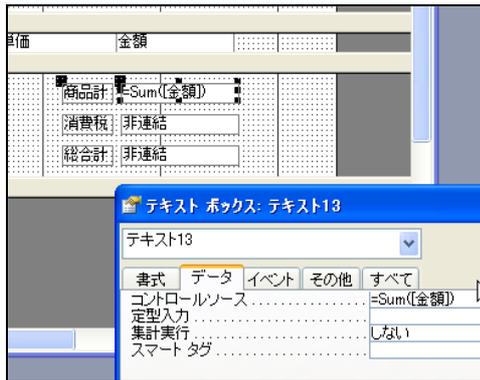
消費税を表示するためのテキストボックスに次の式を登録する。

式: =Fix([A]\*0.05)

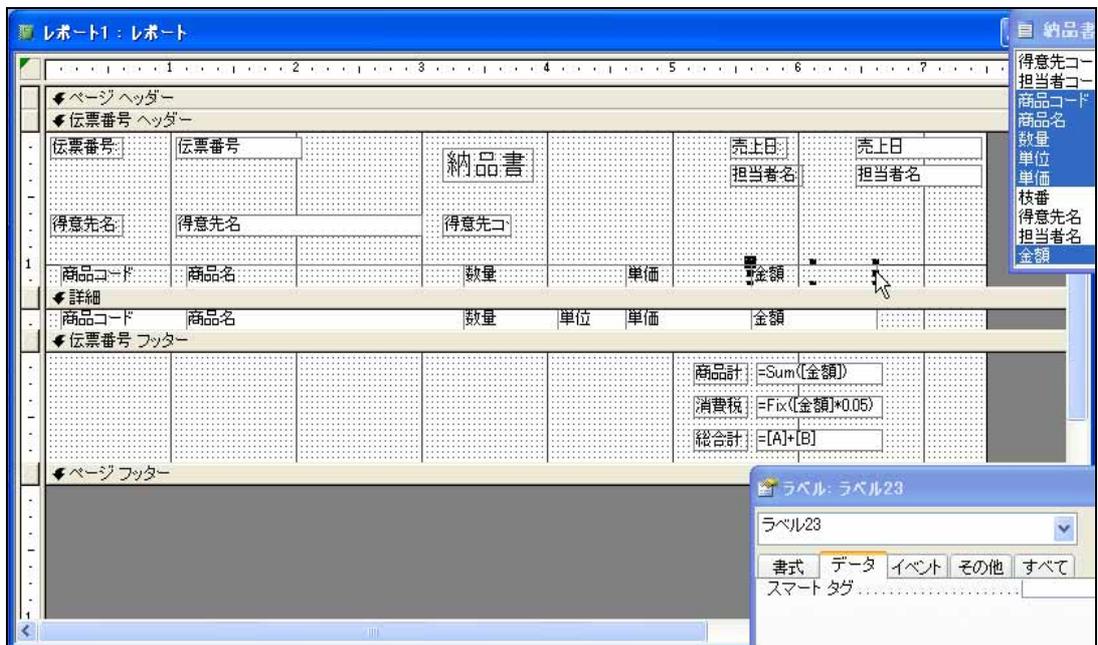
テキストボックスの名前を『B』に変更する。

総合計を表示するためのテキストボックスに次の式を登録する。

式：=[A]+[B]

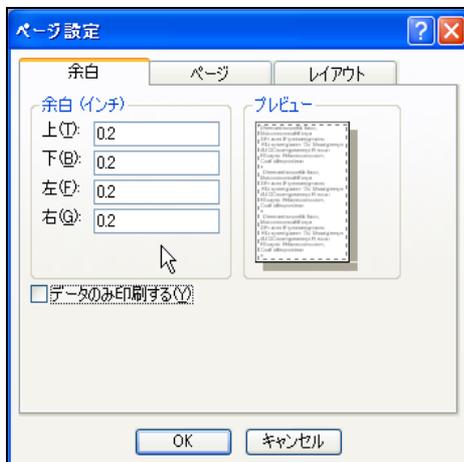


26. 売上明細として必要なラベルを登録する。



27. ここでレポートの用紙サイズ、余白を確認する。

この作業はレポート作成の一番最初に行うのが一般的だが、今回は特別に後回しにした。ここでは余白を最小値にする。



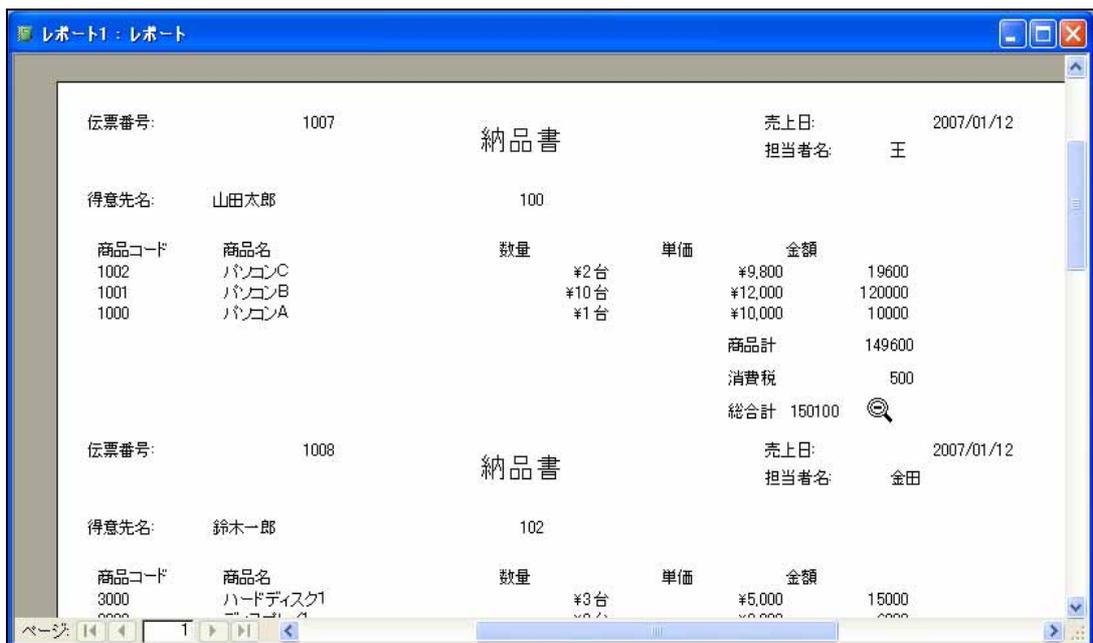
28. 用紙サイズと使用するプリンターなどを確認する。  
ここでは A4 サイズを指定する。



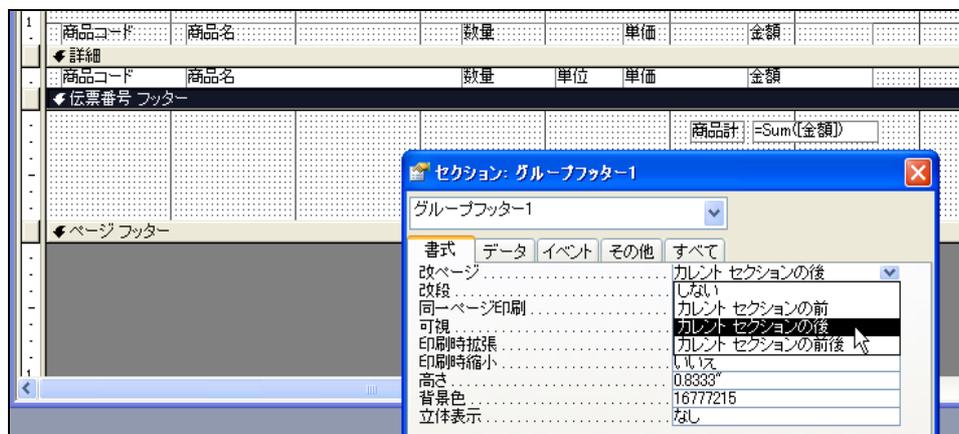
29. ここまでのところで、レポートをプレビュー表示して見よう。  
レポートをプレビュー表示するには画面左上の[ビュー]アイコンをクリックする。



30. レポートが表示される。



31. よく見ると、1 ページに 2 枚の伝票が続けて表示されていることがわかる。  
これを 1 ページに 1 伝票ずつ印刷する設定に変更する。  
[伝票番号フッター]部のプロパティを表示し、[改ページ]を[カレントセクションの後]に設定する。



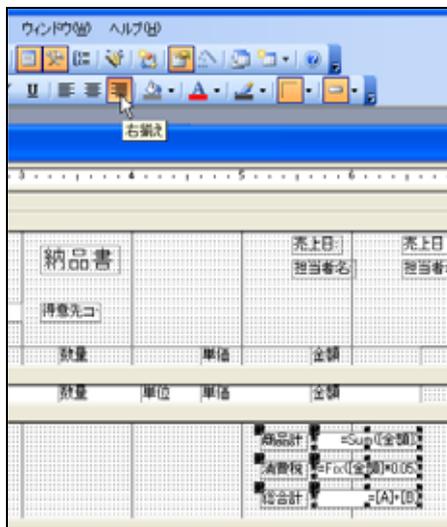
32. レポートをプレビュー表示して確認する。



33. あとは細かな設定となる。

金額を表示するテキストボックスの書式設定を変更し、文字位置なども右揃えに統一する。  
書式設定はフォームの作成時にも登場した次の書式を利用する。

書式：#,##0.#



34. 再度レポートをプレビュー表示して画面を確認する。



35. 最後にレポートに名前を付けて保存する。

以上