

表計算ソフトを活用した簡易データベースシステムの開発に関する研究

三 沢 英 貴¹

Hidetaka MISAWA : A Study on a Development of a Simple Database System Using Spreadsheet Software

現代の代表的なデータベースであるリレーショナル型データベースは、SQL 言語を用いて管理されることが一般的である。しかしながら、SQL を用いたデータベースの管理は専門的な知識と技術を必要とすることから外部の専門家に発注することも多い。そこで、本稿では日常的な業務に幅広く利用されており、専門的な知識と技術についても比較的触れやすい代表的な表計算ソフトである Microsoft 社の Excel とその拡張機能である VBA を活用した簡易データベースシステムの開発と実装の考え方について述べる。

キーワード：データベース Excel VBA

はじめに

データベースのモデルは、階層型とネットワーク型とリレーショナル型の三種に大別される。階層型データベースは、データの親子関係のルートを記録した状態でデータを管理するタイプ（複数の親子関係は重複データで管理）であり、組織図等に代表される構造である。ネットワーク型データベースは、階層型モデルと近い構造を持つが、重複データを持たずデータを管理するタイプ（親子関係については、各データを参照する形で管理）であり、階層型と比較して効率的ではあるが管理が複雑となる。リレーショナル型データベースは、表形式でデータを管理するタイプであり、親子関係を含んだデータの関係については複数の表を関連付けることで実現させる特徴を持つ¹⁾。

リレーショナル型データベースは、複数の表でデータを管理することから、複雑なデータの管理もイメージしやすく、現代における代表的なモデルである。他方、その管理においてはデータベース専用

言語である SQL (Structured Query Language) を用いるため、専門的な知識と技術を要することから外部の専門家に発注することも多い。そこで、本稿では日常的な業務に幅広く利用されており、専門的な知識と技術についても比較的触れやすい代表的な表計算ソフトである Excel (Microsoft Excel2013～2019 及び Microsoft 365：以下 Excel) とその拡張機能である VBA (Visual Basic for Applications：以下 VBA) を活用した簡易データベースシステムの開発と実装の考え方について述べる^{2),3)}。

Excel は、直感的にデータを表形式で管理することに優れているため、データの一元管理が容易であり、拡張機能である VBA と連動させることで入力フォーム、入力ミス防止のエラーチェック機能、データベースにおける主キー条件に必要な一意性の担保などが可能となる。

1. 簡易データベースシステムの構造

本稿では、自己紹介データベースを例に簡易データベースシステムの開発と実装について論述を展開する。自己紹介データベースは、鳥取短期大学 生活学科 情報・経営専攻 (以下、本専攻) 2 年次

1 鳥取短期大学生活学科

自己紹介データベース											
機能選択 ⇒ <input type="button" value="データ入力"/> <input type="button" value="データ修正"/>											
主キー	氏名	出身	年齢	趣味1	趣味2	好きなこと(もの)	嫌いなこと(もの)	得意科目(分野)	苦手科目	卒業後	目指す仕事
1	管理行	管理行	管理行	管理行	管理行	管理行	管理行	管理行	管理行	管理行	管理行
2	三沢英貴	大阪府	43	漫画(読み専門)	妄想	さまぁ〜ず	梅干し	経営工学	世界史	その他	研究職
10000	判定行	判定行	判定行	判定行	判定行	判定行	判定行	判定行	判定行	判定行	判定行

図1 自己紹介用DBシート

後期科目である「データベース」および同2年次通年科目である「特別研究」において、Excelを活用した簡易データベースシステムの開発と実装の基本的な考え方を理解するために開発したものである。

自己紹介データベースは、自己紹介用DBシート(図1)、各種バックデータシート、自己紹介用テンプレシート(図2)の三種類のシートから成る構造を持つ(図3)。自己紹介用DBシートは、主キー(データベースにおける行(カラム)の一意性を保証する役割)から目指す仕事までの12項目を一元管理する役割を持っており、先頭行(主キー番号が1の行)および最終行は、行数を管理するための行である。各種バックデータシートは、自己紹介用DBシートへデータ(レコード)を入力する際に補

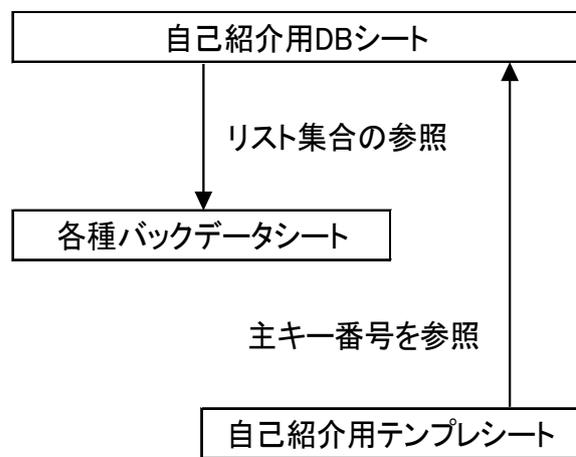


図3 自己紹介データベースの構造

助機能として設定するリストデータの集合を管理する役割を持つ。自己紹介用テンプレシートは、自己紹介用DBシートの主キー番号で管理されている各行のデータを抽出する役割を持つ。自己紹介データベースは、本専攻の学生が理解しやすいようなテーマとしているが、本システムの考え方を一般企業に置き換えることで社員管理、顧客管理、受発注管理などの様々な場面にて応用可能である。

2. 入力機能の実装

(1) ユーザーフォーム機能の実装

自己紹介DBシートへのデータ入力は、入力者の負担を考慮してユーザーフォームを活用するため、その実装について述べる。フォーム機能自体はExcelのオプション機能に存在しているが、データベースにおける各行の一意性の保証、新たに開発し

自己紹介		
		データ読込
		リセット
2021/3/28		
氏名	_____	
出身地・年齢	_____	
趣味	① _____	② _____
好きなもの	好きなもの(こと)	嫌いなもの(こと)
嫌いなもの		
科目について	好きな(得意な)科目	嫌いな(苦手な)科目
卒業後は?	_____	
目指す職業は?	_____	

図2 自己紹介用テンプレシート

た機能と連動が困難であるため、独自設計を行った。Excelの開発タブ(多くの場合、Excelの初期設定では表示されていないため、オプションにて表示設定が必要)からVisual Basic(以下VB)を選択、ユーザーフォームをクリックすることで開発画面にてユーザーフォームを設計することが可能となる。本稿では、図1にあるような全12項目をユーザーフォームへ設定した(図4)。その後、図1にあるデータ入力ボタン(開発タブの挿入からボタンを選択で設定可能)へユーザーフォームの呼び出し命令を登録する(対象のボタンを右クリックからマクロの登録で可能)。VBによる呼び出し命令は図5の通りである。ここで、図5の『UserForm1』は設計したユーザーフォームのオブジェクト名であり、『Show』は指定したオブジェクトを呼び出すための命令である。

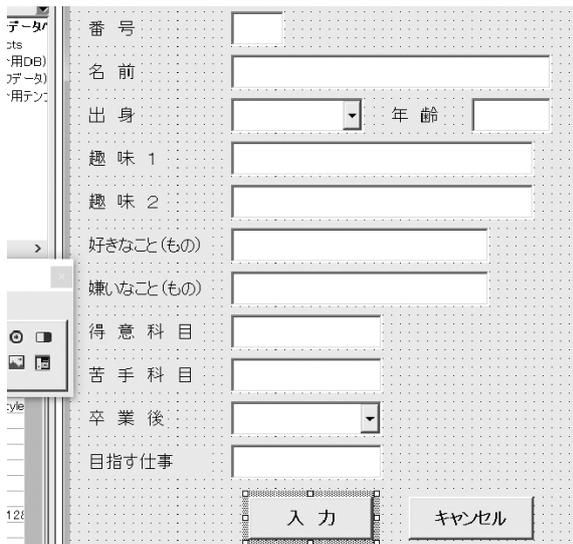


図4 ユーザーフォーム

```
Sub 入力フォーム呼び出し()
    UserForm1.Show
End Sub
```

図5 ユーザーフォーム呼び出し命令

(2) ユーザーフォームを用いた入力

前節において実装したユーザーフォームを用いて自己紹介DBシートへのデータ入力を行う。具体的には、自己紹介DBシートへ設定したデータ入力ボタン(図1の上部左側)をクリックすることでユーザー

フォームを呼び出した後、データを入力してユーザーフォームの入力ボタンをクリックすることで自己紹介DBシートへのデータ転記が完了する(図6、図7)。図7から自己紹介DBシートへ自動的に新たな行が追加されてデータが転記されていることが分かる。つまり、データ転記のための行を最初から確保している訳ではなく、データ転記のタイミングで行を追加した後に転記する仕組みをユーザーフォーム側に実装している。

ここからは、データ転記命令について述べる。データ転記命令は、ユーザーフォームの入力ボタンに登録されており、図8がデータ転記命令である。ここで図8における『With』は、以降に記述される名称の範囲を処理対象として扱う命令、『Rows.Count』は行数をカウントする命令、『EntireRow.Insert』は行を追加する命令である。つまり、これらを組み合わせることで対象の範囲(本稿では自己紹介用DBシート)の有効行をカウントした後に行を追加する処理を実現している。また、同図における『TextBoxbangou.Text(Valueでも可)』はユーザーフォームにおける各種テキストボックスのオブジェクト名であり、対象のテキストボックス内のテキスト(値)という意味を持つ。最後の『Unload



図6 ユーザーフォームへのデータ入力例

主キー	氏名	出身	年齢	趣味1	趣味2	好きなこと(もの)	嫌いなこと(もの)	得意科目(分類)	苦手科目
1	管理行	管理行	管理行	管理行	管理行	管理行	管理行	管理行	管理行
2	三沢英典	大阪府	40	漫画(読み専門)	安眠	ままへず	斬り	経営工学	世界史
3	鳥取短大	鳥取県	50	研究・教育	地域貢献	0000	△△△	◇◇◇	□□□
1000	判定行	判定行	判定行	判定行	判定行	判定行	判定行	判定行	判定行

図7 自己紹介DBシートへのデータ転記例

```
Private Sub CBN_Click()
    'カウント用変数の確保
    Dim i As Integer
    '行数確保用変数の確保
    Dim rowCount As Integer
    'Withステートメントの利用
    With Worksheets("自己紹介用DB").Range("自己紹介データベース")
        '行数確保
        rowCount = .Rows.Count
        '最下行を追加
        .Cells(rowCount, 1).EntireRow.Insert
        'データをセルへ入力
        .Cells(rowCount, 1) = TextBoxbangou.Text
        .Cells(rowCount, 2) = TextBoxname.Text
        .Cells(rowCount, 3) = ComboBoxfrom.Text
        .Cells(rowCount, 4) = TextBoxage.Text
        .Cells(rowCount, 5) = TextBoxhoby1.Text
        .Cells(rowCount, 6) = TextBoxhoby2.Text
        .Cells(rowCount, 7) = TextBoxlike.Text
        .Cells(rowCount, 8) = TextBoxhate.Text
        .Cells(rowCount, 9) = TextBoxtokui.Text
        .Cells(rowCount, 10) = TextBoxnigate.Text
        .Cells(rowCount, 11) = ComboBoxsinnro.Text
        .Cells(rowCount, 12) = TextBoxjob.Text
    End With
    '入力終了
    Unload Me
End Sub
```

図 8 データ転記命令

Me』は、データ転記後に自分自身（ユーザーフォーム自身）を閉じるための命令である。

(3) データの修正

自己紹介 DB シートへデータを転記後、データの修正を行う場合は、データ修正ボタン（図 1 の上部右側）をクリックすることで呼び出される修正番号入力フォームに修正対象データの主キー番号を入力することで可能となる（図 9）。その後は自己紹介 DB シートの主キー番号のデータがユーザーフォームへ転記されることで修正を実行する。ユーザーフォームへの転記命令を図 10 に示す。命令の考え



図 9 修正番号入力フォーム (3 を選択)

```
Private Sub CBK_Click()
    Dim 修正 As Integer '修正番号格納用変数
    修正 = TextBoxrebangou.Text
    'withステートメント利用+データをユーザーフォームへ
    With Worksheets("自己紹介用DB").Range("自己紹介データベース")
        UserForm2.TextBoxbangou2.Text = 修正
        UserForm2.TextBoxname2.Text = .Cells(check, 2)
        UserForm2.ComboBoxfrom2.Text = .Cells(check, 3)
        UserForm2.TextBoxage2.Text = .Cells(check, 4)
        UserForm2.TextBoxhoby12.Text = .Cells(check, 5)
        UserForm2.TextBoxhoby22.Text = .Cells(check, 6)
        UserForm2.TextBoxlike2.Text = .Cells(check, 7)
        UserForm2.TextBoxhate2.Text = .Cells(check, 8)
        UserForm2.TextBoxtokui2.Text = .Cells(check, 9)
        UserForm2.TextBoxnigate2.Text = .Cells(check, 10)
        UserForm2.ComboBoxsinnro2.Text = .Cells(check, 11)
        UserForm2.TextBoxjob2.Text = .Cells(check, 12)
    End With
    UserForm2.Show '修正用フォーム開く
    Unload Me '自身を閉じる
End Sub
```

図 10 ユーザーフォームへの転記命令

方自体は図 8 に示した命令の逆処理であること、ユーザーフォームへの転記後は、図 8 に示す命令と同様の処理を行うだけであるため、説明は省く。

3. データ抽出

リレーショナル型のデータベースにおける代表的な抽出は、必要な項目（列（フィールド））を選択して抽出、表（テーブル）同士を結合して必要な部分のみを抽出などが挙げられる。しかしながら、Excel には、行列の非表示機能やフィルター機能などが存在しており、それらの活用で十分である。そこで本稿では、自己紹介 DB シートに入力されたデータを自己紹介用テンプレシートへ抽出する仕組みについて述べる。

自己紹介用テンプレシートは、自己紹介 DB シートの第一項目（図 1 の主キー）によって管理されている各行のデータを抽出する役割を持つ。具体的には、自己紹介用テンプレシートのデータ読込ボタンをクリックすることで呼び出されるデータ読込番号入力フォームに抽出対象データの主キー番号を入力することにより抽出が完了する（図 11, 図 12）。データ読込番号入力フォームには、抽出対象データの主

図 11 データ読込番号入力フォーム (3 を選択)

```
Private Sub CBR_Click()
    Dim 番号 As Long '読込番号確保用
    番号 = TextBoxread.Text
    'データ読込
    Worksheets("自己紹介用テンプレ").Range("A5") = TextBoxread.Text
    Unload Me '自身を閉じる
End Sub
```

図 13 主キー番号確保命令

図 14 抽出データのリセット

図 12 自己紹介用テンプレ (抽出後)

```
Sub リセット()
    Dim rc As Integer
    rc = MsgBox("抽出データをリセットします", vbYesNo, "データリセット確認")
    If check = vbYes Then
        Worksheets("自己紹介用テンプレ").Range("A5") = ""
    Else
        End If
End Sub
```

図 15 抽出データのリセット命令

4. 一意性の保証とエラーチェック機能

キー番号のみを確保する命令のみが登録されている (図 13)。この理由は、自己紹介用テンプレシート内の各項目を抽出するセル内に自己紹介 DB シートの主キー番号を参照するための命令が登録されているためである。具体的には、Excel の標準的な関数である『VLOOKUP』を活用した命令が登録されており、図 13 の命令と組み合わせることで抽出を実現させている。また、自己紹介用テンプレシートの右上部にあるリセットボタンをクリックすることで抽出したデータを同シートから削除することが可能である (図 14, 図 15)。ここで、図 15 における『MsgBox』はユーザーに対して処理内容の確認を促す命令であり、条件分岐命令である『If』と組み合わせることでリセット処理の確認を実現している。

前章まで、自己紹介データベースの代表的な機能 (入力と抽出) を中心に述べてきた。本章では、それらの機能を安定的に活用するための各種チェック機能 (主キー条件である一意性の保証や入力ミス防止機能) の実装について述べる。

(1) 一意性の保証

リレーショナルデータベースにおける主キーの重要な役割は、表形式で管理しているデータにおいて行単位で一意性を保証するものである。主キーには、一意性を保証するための禁則処理が二つ存在する。一つ目は Null (空白状態) の禁止であり、二つ目は値の重複の禁止である。自己紹介データベースでは、ユーザーフォームへのデータ入力後に自己紹介 DB シートへ転記を行うため、データの転記の

タイミングで空白状態および値の重複を禁止する機能を組み込む必要がある。具体的には、図8に示した自己紹介DBシートへのデータ転記命令に図16の命令を組み込むことで空白状態を禁止することが可能となる。図16の命令は『If』と『MsgBox』と『SetFocus』(カーソルを戻す命令)を組み合わせたものであり、ユーザーフォームへ主キー番号が入力されなかった場合、転記処理が実行されずに主キー番号の再入力を要求する命令である(図17)。

他方、値の重複を禁止する処理については、図16の命令に続ける形で加えていく必要がある(図18)。

図18の命令は、『For』と『For Each』のループ命令二種と『If』の組み合わせにより、ユーザーフォームに入力した主キー番号と自己紹介DBシート内の既存の主キー番号を比較、同一の値と判断された場合は主キー番号の再入力を要求する命令である(図19)。参考までに一意性の保証を有するデータ転記命令を図20へ示す。また、同様の処理が必要

```

If TextBoxangou.Text = "" Then
    MsgBox "管理番号を入力して下さい", vbExclamation, "必須入力エラー"
    TextBoxangou.SetFocus
Exit Sub
End If
    
```

図16 空白状態を禁止する命令

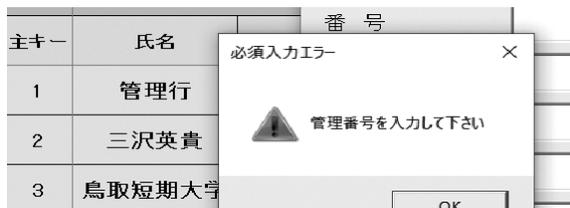


図17 主キー番号の再入力要求処理(空白時)

```

.....
For i = 1 To rowCount - 1
    For Each checkprimarykey In .Cells(i, 1)
        If checkprimarykey.Value = TextBoxangou.Text Then
            MsgBox "主キーの重複は禁止されています!"
            cancel = True
            Exit Sub
        End If
    Next checkprimarykey
Next i
    
```

図18 値の重複を禁止する命令



図19 主キー番号の再入力要求処理(重複時)

```

Private Sub CBN_Click()
    'カウント用変数の確保
    Dim i As Integer
    '行数確保用変数の確保
    Dim rowCount As Integer
    '主キー重複確認用の範囲変数を確保
    Dim checkprimarykey As Range
    If TextBoxangou.Text = "" Then
        MsgBox "管理番号を入力して下さい", vbExclamation, "必須入力エラー"
        TextBoxangou.SetFocus
        Exit Sub
    End If
    'Withステートメントの利用
    With Worksheets("自己紹介用DB").Range("自己紹介データベース")
        '行数確保
        rowCount = .Rows.Count
        '行数1つ上までの範囲内(既確定のデータベース範囲)までをTextBoxに
        For i = 1 To rowCount - 1
            For Each checkprimarykey In .Cells(i, 1)
                If checkprimarykey.Value = TextBoxangou.Text Then
                    MsgBox "主キーの重複は禁止されています!"
                    cancel = True
                    Exit Sub
                End If
            Next checkprimarykey
        Next i
        '最下行を追加
        .Cells(rowCount, 1).EntireRow.Insert
        'データをセルへ入力
        .Cells(rowCount, 1) = TextBoxangou.Text
        .Cells(rowCount, 2) = TextBoxname.Text
        .Cells(rowCount, 3) = ComboBoxfrom.Text
        .Cells(rowCount, 4) = TextBoxage.Text
        .Cells(rowCount, 5) = TextBoxhoby1.Text
        .Cells(rowCount, 6) = TextBoxhoby2.Text
        .Cells(rowCount, 7) = TextBoxlike.Text
        .Cells(rowCount, 8) = TextBoxhate.Text
        .Cells(rowCount, 9) = TextBoxtokui.Text
        .Cells(rowCount, 10) = TextBoxnigate.Text
        .Cells(rowCount, 11) = ComboBoxinnro.Text
        .Cells(rowCount, 12) = TextBoxjob.Text
    End With
    '入力終了
    Unload Me
End Sub
    
```

図20 データ転記命令(一意性の保証有)

な部分(自己紹介DBシートの修正用フォームや自己紹介用テンプレートのデータ番号読込フォーム)についても同様の命令を加えることで、データ転記時やデータ抽出時の人的ミス防止している。

(2) 簡易エラーチェック機能

一意性の保証の観点以外でも入力時の入力ミスが起こりうる可能性がある。もっとも可能性が高いと考えられるミスは、必須入力内容が未入力のまま処

理されることだろう。自己紹介データベースにおける対応策としては、二通り考えることが出来る。一つは、前節の考え方を活用することで転記時に全ての必須入力内容について、再入力要求処理命令を加える方法である。もう一つは、自己紹介用テンプレシートへのデータ抽出時に利用者へ必須入力項目であることを知らせる方法である。本来は、前者の方法を推奨するが、ここでは簡易エラーチェックの紹介も含める意味も込めて後者の方法で実装した。具体的には、Excelの標準的な関数である『If』と『VLOOKUP』を組み合わせることで実現が可能となる。自己紹介用テンプレシートにデータを抽出する際の参照先である自己紹介DBシート側にデータが入っていないければ、自己紹介用テンプレシート側にてエラー表記を示す命令を付け加えた(図21)。



図21 簡易エラーチェック例

おわりに

本稿では、表計算ソフトを活用した簡易データベースシステムの開発と実装の基本的な考え方につ

いて述べた。SQLによる専門的な知識と技術を用いたデータベース管理が非常に重要である一方、Excelのように日常的に利用することが多いツールを活用して、簡易的にデータベース管理を実現させる考え方も重要である。本稿で示した考え方は、顧客管理、社員管理、見積管理、受注管理などへの応用も容易であり、またデータベース管理以外にも表計算ソフトの応用として有意義であると考えている。その反面、SQLのようにある特定の専門領域に特化した技術と完全に同様な機能を持たせた開発は困難であるため、柔軟に活用していく必要があることを付け加えておく。本専攻の卒業生は表計算ソフトの応用的な活用を求められることも多く、また著者の専門分野(経営工学)上、企業から表計算ソフトの応用に関する相談が寄せられることも多い。今後も担当科目やゼミの指導などにおいて、その経験を活かした教育を心掛けていきたい。

引用・参考文献

- 1) 甲木洋介「データベースの初歩と、現在の活用例」、『知能と情報(日本知能情報ファジィ学会誌)』Vol 24 No 3 (2012), pp. 100-101.
- 2) 日本マイクロソフトオフィシャルホームページ (Microsoft-365 Excel), <https://www.microsoft.com/ja-jp/microsoft-365/excel> (2021.03.31).
- 3) 日本マイクロソフト Excel VBA リファレンス, <https://docs.microsoft.com/ja-jp/office/vba/api/overview/excel> (2021.03.31).