

Access と Excel を連携した科学館資料データベースの作成

江越 航*

概要

大阪市立科学館では、現在約 15,000 点の資料を保有している。収集した資料のデータは紙ベースで登録・管理しているほか、登録内容を Microsoft Excel にも入力して電子データにしている。しかし Excel は登録データの検索や分類・更新など、データベースとしての利用を行うには効率的でない。そこでデータベース本体を Access で作成し、インターフェースとして Excel の VBA を利用して Access のファイルに接続するプログラムを作成した。本稿では Access のデータベースを、Excel と連携して活用するための方法について述べる。

1. はじめに

大阪市立科学館では、現在約 15,000 点の資料を保有している。

博物館の基本活動として、資料の収集・保存が挙げられる通り、資料は博物館活動の基本である。科学館でも、各学芸員が日々資料の収集を行っており、登録に至る件数としては、毎年、数十点ずつの資料収集を継続している。

館内にて登録が認められた資料のデータは、紙ベースの資料受入票を作成して登録・管理しており、登録内容を Microsoft Excel のファイルにも入力して電子データにしている。しかし Excel を使用して、登録データの検索や分類、更新等の管理を行うことは効率的でない。資料をデータベースとして管理するには、同じく Microsoft 社から販売されているデータベースソフト Access が適しているため、こちらのソフトでデータベースを作成した。しかし Access は、一般的に販売されているパソコンのソフトウェアの中にはあらかじめインストールされておらず、職員間で共有して使うことが難しい。そこでデータベース本体は Access で作成したファイルを利用しつつ、インターフェースとして Excel の VBA を利用して Access のファイルに接続することを試みた。

本稿では今回作成した Access のデータベースを、Excel と連携して活用するための方法について述べる。

2. Access によるデータベースの作成

2-1. Excel のデータ

Excel で管理しているデータは、図 1 のような形式になっている。1 件の資料受入票が 1 行のデータに対応しており、それぞれの列が資料名や受入日、資料点数などの項目に対応している。今回のデータベースの作成は、この Excel データをもとに行った。



図 1 Excel のデータベース

2-2. データのインポート

Excel で作成したファイルを Access データベースに変換するには、Access のメニューから[外部データ]-[Excel]とすることで、データをインポートすることが可能である。

この際、データベースの各フィールドのデータ型をあらかじめ指定してインポートを行うが、数値型のデータ型に指定したフィールドに文字型のデータが含まれていると、インポートエラーとなってしまふ。

また、テキスト型に指定したフィールドに関しても、後

*大阪市立科学館学芸課
e-mail:egoshi@sci-museum.jp

ほど正規化作業を行う際に、表記方法の違いが問題となる。

例えば、場所の項目として「1 階」と「1F」などの表記の揺れや、文字種として全角と半角の違いがあると、正規化の際には別の項目として扱われることになる。さらに、資料の保管場所に関しては、入力した担当者により独自の名前が付けられ、表記がバラバラであった。

これらの表記方法の違いは、Access で修正作業を行うよりインポート前に統一しておく方が、後の作業が簡単になる。今回のデータベース作成では、この統一作業に多くの時間を費やす必要があった。

2-3. 正規化

Access は、リレーショナルデータベース(関係データベース)と呼ばれるデータベースで、データベースとしては最も普及している形式のものである。

リレーショナルデータベースとは、「関係モデル」という概念に基づきデータを扱うデータベースである。データの管理を複数の表形式のテーブルを用いて行い、それぞれの表を互いに関連付けたデータベースとなっている。

データベースでは、同じデータが繰り返し現れることが多い。この重複するデータはまとめて管理しておかないと、データの修正があった際に、一部分だけ修正してしまうなどして、データベース全体の整合性がとれなくなってしまう。そこで重複するデータを別テーブルに分離することで、重複をなくして整合的にデータを扱えるようになる。この作業を正規化と呼んでいる。

今回、データベースの正規化作業にあたり、次の項目に関するテーブルを作成した。

分野	受入	種類	市/財団
受入者	階数	所在部屋	

また、今後の資料管理状況を把握するために、以下の項目に関するテーブルも追加することにした。

登録抹消	所在確認
------	------

2-4. 資料保管場所の記述

今回整備したデータベースの項目の中で、特にデータベースの構造を考慮した項目として、資料保管場所が挙げられる。資料は保管している部屋に応じて、ロッカーであったりスライドラックであったりいろいろな場所に置かれ、階層が下層になるにつれ、さまざまな分類に分かれる。このような保管場所を示す際には、図2のような階層型形式にできれば記述が容易である。

しかし、Access はリレーショナルデータベースであり、上記のような階層型のデータベースの記述には向いて

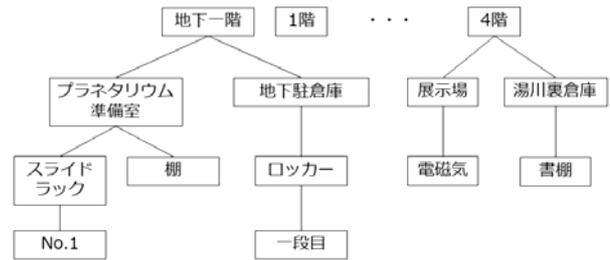


図2 階層型のデータベース

いない。

リレーショナルデータベースにおいて、上記の構造をそのまま記述しようとするれば、一番下層の項目を基準にすべての保管場所に ID を割り振る必要があり、保管場所の項目が膨大になる。また、各保管場所の最下層までの階層が統一できないため、整合性のあるテーブルを作成するのも難しい。

そこで今回は、すべての保管場所に ID を付けることはせずに、約 40 か所の所在部屋に関してのみ、ID を割り振った。そして、それ以上の細かい保管場所に関しては、資料 ID に紐づけられた所在場所の項目に、コメントとして記載することとした。

しかし、登録や検索の際に 40 か所の選択肢から選ぶのは、まだ選択肢数として多い。入力の際、部屋の階数と組み合わせられると便利なので、所在部屋のテーブルは、階数の ID を含めたものを作成した。

所在部屋ID	階数ID	所在部屋	クリックして
1	1	プラ準備室	
2	1	オムニ映写室	
3	1	オムニ映写室	
4	1	プラホール東ノ	
5	1	地下駐倉庫	
6	1	ホワイエツアイ	
7	1	アトリウム	
8	1	事務室	
9	1	館長室	
10	1	書庫	
11	1	工作室	

図3 所在部屋のテーブル

2-5. フォームの作成

正規化作業で分割したテーブル同士にはリレーションを設定し、データベース全体の構築を行った。

さらに、入力用のフォームを作成することで、データ入力がやり易いようにした。

入力用のフォームとしては、資料の全体が一覧で表示されるフォームと、各々の資料の詳細データが表示されるフォームの 2 種類を準備している。

フォームで入力する際、定型的な項目は入力用のコンボボックスを用いてあらかじめ入力可能な選択肢を用意しておき、この中から選ぶ形にした。これは入力の手間を省くとともに、表記の揺れがないようにするた

めである。

詳細データ表示フォームにおいては、表示フォーム中に独立した所在確認の項目を作った。これは資料の現状を分かりやすく表示することで、資料管理に役立てるためである。



図4 詳細データ表示フォーム

2-6. コンボボックスによる絞り込み

一覧表示フォームでは、必要な資料を見つけやすいように、次の要領で、取得年や保管場所、資料名等でデータを絞り込む機能を設けた。

一覧表示フォーム内に、絞り込み条件を入力するテキストボックス、もしくはコンボボックスを設けておく。

一方、このフォームのデータの表示元であるクエリにおいて、例えば次のような条件をつけた新規フィールドを作成する。

```
IIf([Forms]![F_全体一覧]![開始年_テキスト],IIf([T_全体]![年]>=[Forms]![F_全体一覧]![開始年_テキスト],True,False),True)
```

すると、フォームの開始年のテキストフィールドに値がある場合、その値よりもデータベース中の開始年が大きいと値が True となって、必要なデータが抽出される。また、開始年のテキストフィールドに何も値がない場合も値が True となるので、この場合はすべてのデータが抽出されることになる。

上記のデータ抽出が、フォームに条件を入力した際に反映されるためには、再クエリが必要である。そこで、再クエリのみを行うマクロを用意し、絞り込み条件用のテキストボックス、またはコンボボックスのプロパティにおける更新後処理のイベントとして、再クエリのマクロを実



図5 一覧表示フォームと絞り込み用コントロール

行するようにする。すると、フォーカスが別のコントロールに移動したタイミングで再クエリが実行され、データの絞り込みが行われる。

2-7. コンボボックスの連動

2-5 でフォームを作成したが、コンボボックスで所在部屋を選択する際に、選択肢が 40 個も存在するため、この中から目的の場所を選んで入力するのは不便である。

そこで、先に部屋の階数をコンボボックスで選択すると、それに応じて所在部屋の候補を絞り込んで表示できれば便利である。そのための機能を、以下の要領で実装した。

2-4 で述べた通り、所在部屋のテーブルには、階数の ID も含まれている。そこでフォーム中の所在部屋コンボボックスのプロパティにおいて、値集合ソースを設定し、その中で階数 ID の抽出条件に[Forms]![F_全体一覧]![階数_コンボ]を加える。これにより、階数のコンボボックスで選んだ値のみが、所在部屋のコンボボックスに設定される。

ただし再クエリを行わないと、階数のコンボボックスで選んだ値が反映されない。そこで、再クエリのみを行うマクロを用意し、階数のコンボボックスのプロパティにおいて、更新後処理のイベントとして再クエリのマクロを実行するようにする。

なお、マクロを使う代わりに、Access VBA で AfterUpdate イベントを利用して再クエリを行うことも可能である。しかし今回の Access データベースにおいては、VBA を用いないようにした。これは、後に修正が必要となった際に困難が生じることを避けるためである。



図6 コンボボックスの連動

その他、必要な機能を追加して、Access を用いた資料管理用のデータベースを作成した。

3. Excel による Access データベースの活用

3-1. Excel の活用

上記で作成した Access によるデータベースを使用することで、資料の追加、検索、更新等の管理を日常的に行うことが、より容易になった。しかし Access は、一般的に販売されているパソコンのソフトウェアにはあらかじめインストールされておらず、作成したデータベースを職員間で共有して使うことが難しい。そこでデータベース本体は上記の Access のファイルを参照しつ

つ、インターフェースとして Excel を利用することを検討した。[1]

3-2 . ADO による Access データベースへの接続

Access データベースを Excel で操作するには、ADO (ActiveX Data Objects) と呼ばれるソフトウェア部品のパッケージを用いることで可能となる。ADO を使用するためには、Excel VBA によるプログラミングが必要となる。Excel VBA を利用するには、[開発] タブの [Visual Basic] を選択して、Excel VBA の開発環境を立ち上げ、必要なプログラミングを行う。

Excel VBA で Access のデータベースに接続するには、Connection オブジェクトを利用する。ADO の Connection オブジェクトは、次のように宣言して作成できる。

```
Dim adoCn As New ADO.DB.Connection
```

Connection オブジェクトを作成したら、Open メソッドにより Access データベースに接続できる。

```
adoCn.Open ConnectionString:=  
"Provider=Microsoft.ACE.OLEDB.12.0;Data  
Source=科学館資料管理.accdb"
```

3-3 . レコードセットの取得

データベースに接続したら、Connection オブジェクトから、Recordset オブジェクトを作成する。Recordset オブジェクトは、指定した Access のテーブルのすべてのレコードを保持したものと考えることができる。

Recordset オブジェクトは、次のように宣言して作成する。

```
Dim adoRs As New ADO.DB.Recordset
```

Recordset オブジェクトを作成したら、データベース中のテーブルを指定して Open メソッドを用いると、指定したテーブルのすべてのレコードが、レコードセットとして取得される。

```
adoRs.Open Source:= "T_全体";  
ActiveConnection:=adoCn, CursorType:=ad  
OpenKeyset, LockType:=adLockOptimistic
```

取得した Recordset オブジェクトを Excel のワークシート上に取り込むには、Excel VBA の Range オブジェクトの CopyFromRecordset メソッドを用いればよい。

```
Range("A1").CopyFromRecordset  
Data:=adoRs
```

このコマンドによりレコードセットに取り込んだテーブルを、A1 セルを左上とする領域に貼り付けることができる。

一行ずつ読み込むには、Recordset オブジェクトの Fields プロパティを用いる。

```
Range("A1").Value = adoRs.Fields("資料名").Value
```

上記の方法で、レコードセットで操作対象となっているカーソル位置にあるレコードから、必要なフィールドの値を読み込みことができる。

レコードセットを配列イメージで読み込むには、GetRows メソッドを用いることができる。二次元の配列変数 arrTable(列,行) を用意しておく、

```
arrTable = adoRs.GetRows
```

とすることで、配列 arrTable から必要な値を取り出すことができる。

3-4 . フォームの作成

前項の手法により、Access のテーブルを Excel のワークシートに一覧表示することが可能となる。

一方、データの検索や更新用には、フォームを作成して入力項目を指定すると便利である。

そこで、最初の立ち上げた画面で、データー一覧とともに、絞り込み用のフォームを表示している。取得年や保管場所、資料名等でデータを絞り込み、必要な資料を見つけたところで資料 ID を入力すれば、その資料に関する詳細データが表示されるようにした。



図7 一覧表示画面と検索用フォーム

3-5 . 検索

上記のフォームでの入力値からデータの検索を行うためには、まずコンボボックス、あるいはテキストボックスにおける Text プロパティを使用して、入力値を取得する。

この値を元に必要なレコードのみを取り出すには、3-3 の Recordset オブジェクトを Open する際に、テーブ

4. おわりに

収集資料の点数が増えてくると、その管理が困難になってくる。博物館資料管理用の専用ソフトも存在するが高価であり、導入に当たってのハードルが高い。

今回、汎用のデータベースソフトである Access を利用し、インターフェースとして Excel の VBA を利用してデータベースとして活用する方法を検討した。

データベースは登録した後も、日々の業務の中で管理の状況が変わり情報が古くなっていくことから、日時的にメンテナンスして情報を更新することが必要である。そのためには館の実情に応じた使いやすさが重

要であるので、それらを考慮してインターフェースを作成した。

さらに、データベースは活用することが重要である。今後館内での管理にとどまらず、web での公開手法についても検討したいと考えている。

参考文献

[1]立山秀利,「Excel VBA で Access を操作するツボとコツがゼッタイにわかる本」, 秀和システム, p248(2015)