

# 音楽ジャンルに合わせた CD オンライン データベースの構築

松田 直子

## 目 次

|   |                             |    |
|---|-----------------------------|----|
| 1 | 序論                          | 1  |
| 1 | 研究の目的 . . . . .             | 1  |
| 2 | 必要な知識 . . . . .             | 2  |
| 2 | 本論                          | 5  |
| 1 | 製作過程 . . . . .              | 5  |
| 2 | ジャンル分けと表現方法 . . . . .       | 7  |
| 3 | データベースの項目 . . . . .         | 8  |
| 4 | 複数テーブル . . . . .            | 13 |
| 5 | 二枚以上セットになった CD . . . . .    | 14 |
| 6 | Excel の表を SQL にする . . . . . | 14 |
| 7 | 各ファイルの説明 . . . . .          | 16 |
| 3 | 結論                          | 19 |
| 1 | アンケート結果と考察 . . . . .        | 19 |
| 2 | 今後の改良点 . . . . .            | 22 |
| 3 | 自己評価 . . . . .              | 22 |

## 1 序論

### (1) 研究の目的

卒業研究のメインテーマは「人の役に立つデジタルツールを作る」である。そこで私は、Unix 上で PHP と MySQL を使用したオンラインデータベースを構築することにした。

音楽を聴くのが好きな私はよく音楽ソフトを利用する。しかし、私のパソコンに最初から入っている音楽ソフト BeatJam<sup>(1)</sup>は CD を入れても曲名を表示せず、自分で曲名などすべて入力しなければならないという要領の悪さがあった。そして、無料でダウンロードできる iTunes<sup>(2)</sup>などの音楽ソフトは、曲名や作曲者名、アルバム名などが CD に記載されている通りに表示されていなかったり、スペルミスや全く違う曲名が入力されている場合も多くある。特にジャンルは種類が少なく、表記されたジャンルが気に入らなければ自分で入力しなければならないという点が気になった。

例えば iTunes で民族音楽の CD を入れると「World」や「Folk」と表記され、クラシック CD を入れると「Classical」や「Unclassifiable (分類できない)」と表記される。これが適切なジャンルではないのは明らかである。このようなジャンル分けになるのはジャンルの選択肢が少ないためではないだろうか。

インターネットや本などでクラシックのような複雑なケースに対応するソフトを探してみても、個人的に作られているもの<sup>(3)</sup>はあるが、ソフトとしてたくさんの人が利用できるものはなかった。そこで私は音楽の中でもジャンル分けが一番難しいクラシックに着目し、クラシック音楽をよく聴く人の役に立つためのジャンルに合わせた音楽データベースを

構築しようと考えた。

これを構築することによって将来的には多様なジャンルのある音楽オンラインデータベースができれば、より快適にパソコンで音楽を聴けるようになり、音楽ソフトを利用する人も増えるようになるのではないだろうか。

## (2) 必要な知識

PHP と MySQL を使用したオンラインデータベースを作るにあたって特に必要な知識を以下で紹介する。

### (i) オンラインデータベースについて

データベースとは、文字や数字といったデータを格納するための入れ物のことである。多くのデータが蓄積されたデータベースは、さまざまな用途で利用される。例えば、商品カタログ、顧客リスト、図書リストなどの情報がデータベースに格納されていれば、必要な情報を素早く取り出すことができ、検索もできる。

また、データベースは文字を格納するだけでなく、必要に応じてデータを取り出すことが可能である。データベースを使ってデータを操作することは、データを「探す」「追加する」「変更する」「削除する」ということである。そして、これらの操作を Web ブラウザさえあればどのコンピュータからでもできるようにしたものが「オンラインデータベース」である。

通常、Access で作られたデータベースは Access からでないと使用できない。また、Excel で作られたデータは Excel のアプリケーションがないと使えない。しかし、オンラインデータベースは、Web ブラウザとネットワーク接続さえあればどのコンピュータからでもデータベース

にアクセスできるように作られている。データベースとデータを Web 上に置くことで、ユーザは Web ページにアクセスする感覚でデータを取り出したり入力したりできるのである。

(ii) MySQL について

データベースを作る際、「データベース管理システム」が必要となる。今回 MySQL を使った理由は以下のことがあげられる。

MySQL はリレーショナルデータベース管理システム (RDBMS) で、マルチスレッド、マルチユーザーの、堅牢で処理速度の速いデータベースサーバーです。また、非常に多くのデータを扱えるため、広い範囲での応用が可能です。代表的な特徴として

- ・ C、C++、Java、Perl、PHP、Python、Rudy などの豊富な API を持つ
- ・ 動的な Web パブリッシングシステムとの接続性に優れている
- ・ 多言語対応で、日本語は SJIS、EUC とともにサポートしている
- ・ DB をまたいでクエリーの発行ができる
- ・ MyODBC によって ODBC をサポートする
- ・ セキュリティシステムの柔軟性が高い
- ・ Windows や各種 UNIX 系 OS など、多くのプラットフォームで動作する

などがあげられます。

(田中ナルミ/阿部忠光 著 『標準 MySQL』 P.20 より抜粋)

このことから、MySQL は基本的に誰もが自由にダウンロードして使用でき、目的に応じてソースコードの中身を修正することが可能であることが分かる。そして、使用方法が簡単な上、処理スピードが速く、Web サーバーとの接続のよさとセキュリティ管理のしやすさが魅力的

である。また、PHP との連帯に優れており、Web 上のシステムでもよく利用されているデータベースシステムということで、今回「MySQL」を使用することにした。

(iii) PHP について

上記で述べたように、PHP は MySQL と最も相性のよいサーバーサイドスクリプトということで今回使用することにしたが、PHP とはどのようなものなのかを以下に述べる。

PHP の名称は、正式には Hypertext Preprocessor といい、HTML に組み込んで使う、サーバーサイドで処理されるスクリプト言語および実行環境です。最近ではレンタルサーバーでも、オプションサービスとして MySQL と PHP を利用できるほどよく使われています。

(田中ナルミ/阿部忠光 著 『標準 MySQL』 P.166 より抜粋)

つまり PHP とは、PHP という名前のプログラミング言語の総称であり、かつ「プログラム」と呼ばれる「コンピュータに対して指令を出すファイル」なので、PHP プログラム自身が HTML 構文の文字列をブラウザに送り返すところである。

(iv) CGI について

PHP ファイルとしてデータベースを公開するのに、なぜ CGI を使うのかを以下に述べる。

ページの内容を動的に変化させ、毎回同じ内容のページではないものを提供するために利用する PHP などのプログラミング言語などのことを、「CGI」と呼びます。変化する外部情報を必要とするページには、PHP などの CGI を利用しないと実現できません。

(紙屋歌寿彦 著 『初めての人のためのかんたん PHP + MySQL

入門』 P.10 より抜粋)

このことから、CGI とは Web ページ上で利用されるプログラミング言語で作られるページの総称であり、掲示板やアクセスカウンタのように、プログラムから外部の変化する情報を取得し、その情報を利用してページの内容が変化するものであることが分かる。

データベースも、追加や更新、削除などユーザーの入力によってページが変化するもので、CGI を使用するものである。

(v) Apache について

本研究では、HTTP サーバである Apache を使用した。

Apache はもともと、NCSA httpd の細かいバグを修正したり新しい機能を追加するためのパッチ (patch) の寄せ集めとして開発されていたため、この名称がついた。現在の Apache は単独で動作する Web サーバとなっており、世界で最も使われている Web サーバとなっている。

(IT 用語辞典 e-Words<sup>(4)</sup>より抜粋)

Apache は、UNIX 系 OS や Windows で動作する高機能な Web サーバである。フリー版でありながら、誰でも修正・再配布することができるのが魅力であり、今もなお活発に開発が続けられていることも長所の一つである。

## 2 本論

### (1) 製作過程

本格的な作品を作るにあたって使用したファイルである。

- ・ music.sql(データベースとテーブル)

- music.html
- ichiran1.php(一覧)
- seach.php(検索)
- insert.php(追加)
- delete.php(削除)
- delete\_d.php(削除確認処理)

これらが出来上がるまでの基礎となったものや問題点をこれから以下に述べていく。

まず初めに『はじめての人のためのかんたん PHP + MySQL 入門』で PHP の基礎を学び、音楽データベースの仕組みをインターネットで調べた。そしてデータベースを作る際にどのような項目が必要か考え、『標準 MySQL』の本を参考にしながら今後の基礎となる簡単なデータベースを作成した。

そのデータベースでは ID、CD 番号、曲名、演奏者の入ったテーブルを一つ作り、一覧、検索、追加、更新、削除が出来るようにした。その時に使用していたファイルは以下のとおりである。

- music.sql(データベースとテーブル)
- music.html
- ichiran1.php(一覧)
- search.php(検索)
- insert.php(追加)
- update.php(更新)
- update\_d.php(更新確認画面)
- delete.php(削除)
- delete\_d.php(削除確認処理)

ID は 1 から順番に入力されるため、music.sql で「ID int auto\_increment primary key,」と入力しておく。

また、テーブルの入力を shift\_jis で入力してしまった場合、knoppix での文字コードは euc\_jp なので変更する必要がある。その場合、「sjis2euc (ファイル名)」と入力すると変更される。逆の場合は「euc2sjis (ファイル名)」である。すべてのファイルを変更したいときは、ファイル名のところに「\*.」と入力する。

基礎となるものを作ることによって、これからすべき事と問題点がはっきりした。問題点は、クラシックのジャンル分けがはっきりしていない点をどうすべきか、表現方法 (作品名、曲名、楽章など。漢字、カタカナ、全角、半角)などをどうすべきかという点である。

## (2) ジャンル分けと表現方法

クラシック音楽のデータベースを作ると決めてから考えていたジャンルは以下の通りである。

- ・ バロック...バッハ、ヘンデル、クーブラン、ヴィヴァルディなど
- ・ 古典派...ハイドン、モーツァルトなど
- ・ ロマン派...ベートーヴェン、シューベルト、ショパン、チャイコフスキーなど
- ・ 印象派...ドビュッシー、ワーグナーなど

しかし、これでは時代で分けてあるだけでジャンルではないことに気がついたので、以下のように変更した。

- ・ 交響曲...四部ほどの楽章からなる (第○番とついていることが多い)
- ・ 管弦楽曲...オーケストラ (第○番とつかずに名前がつくことが多い)
- ・ 弦楽曲...弦楽器のみ

- ・ 協奏曲...オーケストラとソリスト
- ・ 器楽曲...ソリスト
- ・ 声楽曲...オーケストラまたはソリストと歌
- ・ 行進曲
- ・ 舞曲

このように分けることによって、クラシックのジャンルを分かりやすく区別できるようになった。

また、表現方法は CD によって書き方が違うので、輸入版の CD は英語のままでいいのか、または日本語に変えたほうがいいのか考えたが、日本語に変えた場合、訳が的確ではない場合が出てくると正確なデータベースではなくなるので CD に記入されている表現で入力することにした。

### (3) データベースの項目

テーブルを作る際には必ず項目が必要になる。

iTunes の項目は、

- ・ ID
- ・ 名前
- ・ 時間
- ・ アーティスト
- ・ アルバム
- ・ ジャンル
- ・ マイレート
- ・ 再生回数
- ・ 最後に再生した日

となっている。

そして、アーティストの項目に指揮者が入力されていたり、作曲者名は名前 (曲名) のあとに括弧で入力されている。また、ジャンルは「classical」または「Unclassifiable」と表示されることが多い。iTunes で一番問題なのは、CD に書いてある曲名と表示される曲名が違っていたり、輸入版 CD の曲名表記のスペルが間違っていたり、全く違う曲名が入力されていることである。これではクラシック CD を聴く人は iTunes を使うのをやめてしまうだろう。

以上のことを改善するために必要な音楽データベースで必要な項目は、

- ・ 曲 ID
- ・ レコード (CD) 番号
- ・ CD 名
- ・ トラック番号
- ・ 曲名
- ・ 作曲者
- ・ ジャンル
- ・ 指揮者
- ・ 演奏者

であると考えた。

テーブル 1 にはレコード番号、CD 名、作曲者名、テーブル 2 には ID、レコード番号、トラック番号、曲名、作曲者名、ジャンル、指揮者、演奏者に分けてレコード番号でリレーションができるようにした。また、指揮者で集めた CD の場合、テーブル 1 の作曲者名の項目には名前を入力できないので空白にすることにした。

——音楽ジャンルに合わせた CD オンラインデータベースの構築——

『ポップ・クラシック・ベスト』の CD を例に考えてみると、

・ iTunes

名前

1. 01. 愛よ永遠に (モーツァルト)
2. 02. 涙のカノン (パッフェルベル)
3. 四季の春 (ヴィヴァルディ)

以下同様

時間

1. 3:03
2. 3:25
3. 3:14

以下同様

アーティスト レイモン・ルフェーブル

アルバム レイモン・ルフェーブル ポップ・クラシカル・ベスト

ジャンル Unclassifiable

マイレート 「 」 0~5 で表現

再生回数 1~

最後に再生した日 (例)2005/11/25

・ 製作 DB

曲 ID 161~178

レコード番号 11

CD 名 ポップ・クラシック・ベスト

トラック番号 1~18

曲名

1. 愛よ永遠に

2. 涙のカノン

3. 四季の春

以下同様

作曲者

1. モーツァルト

2. パッフェルベル

3. ヴィヴァルディ

以下同様

ジャンル 管弦楽曲

指揮者

演奏者 レイモン・ルフェーヴル・グランド・オーケストラ

ここでの問題点は、iTunes の名前の表示とジャンルである。名前はトラック 1 と 2 にだけ「01.」「02.」といった番号が入っている上、曲名の横に作曲者名が入力されているのも変である。そしてジャンルも全て「Unclassifiable(分類できない)」となっている点も問題である。

次に、輸入版の CD 『BACH SACRED MASTERPIECES』を比較すると、

・ iTunes

名前

1. Chor (Chorus I/II) Kommt, ihr Tochter. helft mir klagen

2. Rezitativ (Evangelist,Jesus) Da Jesus diese Rede vollendet hatte

以下同様

時間

1. 9:52

2. 1:03

以下同様

アーティスト Karl Richter

アルバム J.S.Bach: Matthauss Passion

ジャンル Classical

マイレート 「 」 0～5 で表現

再生回数 1～

最後に再生した日 (例)2005/11/25

・ 製作 DB

曲 ID 676～704

レコード番号 60

CD 名 BACH SACRED MASTERPIECES

トラック番号 1～29

曲名

1. St.Matthew Passion BWV224 Anointing in Bethzany Chorus:  
Kommt, ihr Tochter, helft mir klagen with Chorale: O Lamm  
Gottes, unschuldig
2. St.Matthew Passion BWV224 Anointing in Bethzany Recitative:  
Da Jesus diese Rede vollendet hatte

以下同様

作曲者 J.S. バッハ

ジャンル 声楽曲

指揮者 KARL RICHTER

演奏者 MUNCHENER BACH-CHOR/MUNCHENER BACH-  
ORCHESTER

ここでの問題点は、iTunes では曲名が大分省略されている点である。

そしてアーティストの項目に指揮者の名前が入力されている点についても、作曲者と指揮者として分けた方が見やすいと思われる。

以上を比較して改善した点は、曲名、指揮者、作曲者、演奏者の項目を別々に作り、ジャンルもクラシック用に変更した点である。そして、曲名の入力を CD に書いてある通りに入力することによって、正確さを第一に考え製作した。

#### (4) 複数テーブル

iTunes のデータベースは一つのテーブルで出来ている。しかし、製作したデータベースは CD 名のテーブルと曲のテーブルの二つのテーブルで作ることにした。二つのテーブルにしたメリットは、既存のテーブル間のリレーションシップから、データをよりわかりやすい状態で参照できるようになることである。また、外部キーを「CD 番号 (recnumber)」と設定し、表同士で矛盾のない値を保つことができることである。

たとえば一つの曲を削除したい時、一つのテーブルでできている場合だと一曲分のデータが全て入力されているレコードを削除しないといけないのに対し、二つのテーブルでできていると CD 名を消さずに曲のテーブルに入っている一曲だけを削除すればいいのでわかりやすく管理もしやすいのである。

また、CD 名を間違えて入力してしまった場合、一つのテーブルでできていると、間違った CD 名を入力したレコード全てを変更する必要が生じる。これも二つのテーブルでできていると、CD 名が入っているテーブルの一行を変更するだけでいいのである。

このように一つのデータが複数の場所に分散している構成だと、データベース全体でデータの整合性を保証するのが、規模が大きなデータ

ベースになるほど難しくなるのである。これを回避する手段の一つが「一つのデータは一カ所で集中管理すべき」という RDBMS(リレーショナルデータベース管理システム) の考え方である。

#### (5) 二枚以上セットになった CD

Excel でデータ収集を始め、二枚以上の CD を入力する場合をどうするか迷った。たとえば、「2 - 3」(二枚めの三曲目) の場合、「 - 」の部分をマイナスにすると Excel のシステム上「2 月 3 日」と表示されてしまった。—(ダッシュ) または - (ハイフン) だと正しく表示されたのでハイフンで記入することにした。

#### (6) Excel の表を SQL にする

CD から集めた情報を Excel で入力していたため、music.sql のファイルに移動させる作業が必要となった。

まず、Excel からメニューを開き、「名前をつけて保存」でファイルの種類を(テキストタブ区切り)に選択し、ファイルの名前を変えて保存をする。保存をするときに出てくる質問は全て「はい」を押す。今回はテーブルを二つにしたため、シートも二つ作る必要があった。そのため、シート 1、2 とともに別々にファイルを作った。

そして、シート 1 を music.sql に貼り付ける作業をする。前に作った music.sql ファイルをコピーして新しい music.sql を作成し、INSERT 文『insert into cd1 (recnumber,title,composer) values("』をコピーする。シート 1 をコピーしたファイル(テーブル 1)に戻ってメニューバーの「編集」から「検索」→「置換」を選択し、置換前の欄に「^」(行の最初からという意味)、置換後に先ほどコピーしたINSERTを貼り付け

る。正規表現の欄にチェックを入れて、範囲は「ファイル全体」とする。そしてすべて置換すると『insert into cd1 (recnumber,title,composer) values("1 音楽の捧げもの J.S バッハ』となる。

次に、スペースに「,」を入れないといけないので上記と同じ要領で、置換前に「\t」(スペースの意味)、置換後に「” , ”」を入力する。

行の最後も同じように、置換前に「\$」(行の最後の意味)、置換後に「”);」と入力することで『insert into cd1 (recnumber,title,composer) values("1"," 音楽の捧げもの","J.S バッハ");』という風に表示され、インサート文が完成する。

music.sql ファイルに貼り付ける作業をシート 2 のほうも同じように作成する。そして、インサート文の最初の行に以下を入力して music.sql が完成する。

```
drop database music;
create database music;
use music;
create table cd1 (
    ID int auto_increment primary key,
    recnumber int,
    title varchar(100),
    composer varchar(100)
);
create table cd2 (
    ID int auto_increment primary key,
    recnumber int,
    tracknumber int,
```

```
kyokumei varchar(200),  
composer varchar(100),  
janru varchar(100),  
conduct varchar(100),  
player varchar(100)  
);
```

#### (7) 各ファイルの説明

本製作に使用したファイルの説明である。

##### (i) music.html

ユーザーが Web 上で検索条件を記入したりすることで PHP や CGI を起動するための HTML フォームをここで作っている。見た目はにサンプルで見やすくし、色も目に優しい緑で統一した。

各送信ボタンをクリックすることで、データの「一覧」「検索」「追加」「削除」スクリプトを起動する。

また、input のタグ中にある「name=" "」で記述されている内容は、スクリプト内では変数名として扱われる。そして「value=" "」で記述された内容が代入されたり、ユーザーが記入したものが代入されて処理されるのである。

例えば、データの追加の場合は

```
<form action="insert.php" method="post">  
<tr>  
<td bgcolor="#99FFCC"><input type="submit" value="データの  
追加">  
</td>
```

```
<td>
CD 名<input type="text" name="insert_title" size="40"><br>
トラック番号<input type="text" name="insert_tracknumber"
size="40"><br>
曲名<input type="text" name="insert_kyokumei" size="40"><br>
作曲者<input type="text" name="insert_composer" size="20"><br>
ジャンル<input type="text" name="insert_janru" size="20"><br>
指揮者<input type="text" name="insert_conduct" size="40"><br>
演奏者<input type="text" name="insert_player" size="40"><br>
</td>
</tr>
</form>
```

となる。

(ii) ichiran1.php

このデータベースにはどのようなデータが入っているのか一目で分かるように、一覧表示が見られるスクリプトを作成した。

まず「`$connect = mysql_connect("localhost","root","semi")`」では `mysql_connect()` 関数を使って MySQL に接続する。「接続 ID = `mysql_connect(ホスト名, ユーザー名, パスワード)`」となっている。

そして「`mysql_select_db("music", $connect)`」では `mysql_select_db()` 関数を使って、接続するデータベースを選択する。この場合は「music」というデータベースを使用している。

SQL 文は「`$sql = "selectcd2.ID,cd1.title,cd2.tracknumber,cd2.kyokumei,cd2.composer,cd2.janru,cd2.conduct,cd2.player from cd1,cd2 where cd1.recnumber = cd2.recnumber";`」となっており、リ

レーションシップは cd1 の recnumber と cd2 の recnumber で二つのテーブルを繋いでいる。

( iii ) seach.php

seach.php は、フォームで入力したキーワードを条件として検索を行う PHP である。

「%」(ワイルドカード) を利用することであいまい検索を可能にしている。そのため、全ての記入欄が空欄だった場合は、すべてがワイルドカードとなって、全レコードが表示されることになる。それを避けるために、「if (\$search\_ID == "" && \$search\_recnumber == "" && \$search\_title == "" && \$search\_tracknumber == "" && \$search\_kyokumei == "" && \$search\_composer == "" && \$search\_janru == "" && \$search\_conduct == "" && \$search\_player == "")」を入力することで、すべてが空欄だった場合、キーワードの入力を促すメッセージを表示し、なにかが入っていれば検索をするという流れを作った。

また、キーワードを複数入力すると、and 検索なので、すべてのキーワードを含むレコードが表示される。

( iv ) insert.php

データを追加する insert.php は、フォームに入力されたデータを変数として受け取り、mysql\_query() 関数を使ってそのデータを追加する。

```
$query = "insert into cd1(recnumber,title)
values(\"$insert_recnumber\", \"$insert_title\")";
mysql_query($query);
```

(cd2 も同様にする)

SQL 文中の「"」は、スクリプトの一部として扱われないように、「\"」

を使うことによってプログラム内での機能や役割ではなく、本来の文字として使用することができる。

(v) delete.php、delete\_d.php

削除は、ID をキーとして取得したレコードを消すことができる。delete.php は、レコードを表示するスクリプトであり、delete\_d.php は、実際の削除コマンドを発行するスクリプトである。

delete.php は「`$query = "select * from cd1,cd2 where cd1.recnumber = cd2.recnumber and cd2.ID = \" $delete_ID \";`」で、フォームから取得した ID 番号でテーブルを検索する SQL 文を `$query` へ格納する。そして「`<input type="hidden" name="delete_ID" value="<? = $delete_ID ?>">`」で ID 番号は HTML 内の隠しフォームで送信データとして delete\_d.php に渡される。

### 3 結論

#### (1) アンケート結果と考察

音楽オンラインデータベースが完成したところで、人文情報学科福田洋一ゼミの三回生数名に製作物を実際に使用してもらい、アンケートにご協力頂いた。

アンケートを実施した意図は、第三者から見た客観的な意見を知りたかったからである。どんなに自分が便利で役に立つと考えて作ったプログラムであっても、主観的にしか見ることができない。そのため、いくらユーザーの方のことを考えて作ったとしても、本当に便利で必要なプログラムとは言い難い。そこで第三者に実際に使ってもらうことによって、率直な意見と感想を聞く事が必要であると思いアンケートを実施し

た。アンケートの結果は以下の通りである。

質問 1「項目 (CD 名、トラック番号など) は適切ですか。」での意見には

- ・ 適切でした。
- ・ 作曲者 (ジャンル) ごとに分けられているので適切だと思います。
- ・ 適切だと思うが、他の音楽のジャンルを追加する場合、作詞者、編曲者も必要になってくると思います。

との意見を頂いた。適切だったと言う意見が多かったが、作詞者、編曲者は考えていなかったのも、確かにあれば便利だと思った。

次に質問 2 の「ジャンルは適切ですか。」という問いには、全員が適切であるとの答えであった。

次に質問 3「使いやすい、見やすいですか。」での意見では

- ・ クラシックについてはあまりわからないので、少し見えにくかった。
- ・ 字が多くなってしまうのは仕方ないと思いますが、使いやすく見やすかったです。
- ・ 使いやすいと思います。もう少し大きくしたら見やすくなると思います。
- ・ 検索の所で、一字だけでも検索できればもっといいと思う。(例えば「G 線上のアリア」を「G」のみで検索)

との意見を頂いた。

字が多くなってしまうことで見えにくくなるのが欠点だと気付いた。また、項目が多いことで、画面いっぱいに表示しても狭く感じるのも改良する点だと思った。1 字だけでも検索できるようにとの意見が出たが、これはすでに「%」を使ってあいまい検索ができるようにしてあるので、もう少し使い込んでもらいたかった。

最後に、質問 4「その他、気付いた点がありましたらご記入ください。」には

- ・ データの一覧がちょっとだけ見にくいと感じました。文字を真ん中にしたりするともっと見やすくなると思います。検索とかも、一つの文字を記入するだけで、検索できたりしてすごいと思いました。
- ・ TOP のページがシンプルなのですが、そのままでもいいのでしょうか？
- ・ クラシックをあまり聴かない人には使いにくいと思う。

との意見を頂いた。データの一覧は一番項目が多く、文字も多いので見にくいと感じてしまうのはもっともだと思う。文字を真ん中にするという意見は、自分の中では全く考えていなかったのも新たな発見であった。TOP ページは見やすいようにシンプルにしたつもりでいたが、シンプルすぎたようである。

今回実際に使って頂いた数名は、あまりクラシックを聴く方ではなく知識も少なかったようで、「クラシック音楽をよく聴く人の役に立つためのデータベース」という点では正確な意見ではなかったかもしれないが、第三者の人に使ってもらうことによって、自分では見えていなかった改良点に気付くことができてよかったと思う。また、使って頂いた全員からこのプログラムの意図が分かり、使いやすいと答えてもらったので、自分自身が便利に使えればいいという独りよがりなものではなく、「人の役に立つものを作る」という今回の卒業制作のテーマに沿っていることが実感できた。

## (2) 今後の改良点

データベースを使ってデータを操作するということは、「検索」「追加」「更新」「削除」ということである。最初の目標ではこの全てをできるようにしたかったが、「更新」だけできないままに終わってしまった。

もっとデータベースを良いものにしていくには、CD のデータを増やすことが今後の改良点だと思う。本研究では六十枚の CD を入力したが、これでは枚数が少なすぎて完璧なデータベースとは言い難い。また、ユーザーが曲を追加できるように作成してあるが、一曲ずつ CD 一枚分を追加していくだけでも大変な時間と労力がかかる。追加の方法も改良の余地があると思うが、実際データベースを作成してみて、基礎となるデータの収集がより必要だと感じた。

そしてアンケートを取って気付いた点であるが、シンプルすぎる点と見やすさという点にもっと改良の余地があると感じた。シンプルにすることで見やすさを考えたつもりでいたが、実際フォームはシンプルすぎて面白みに欠けてしまい、一覧表示は文字が多く、見にくいものとなってしまった。改良するならば、シンプルにこだわりすぎずに見やすいものを作りたいと思う。

## (3) 自己評価

今回の卒業研究では「人の役に立つデジタルツールを作る」をメインテーマに作品の製作を進めてきた。人の役に立つものを作るということは、自分では判断しにくい。コンピュータという無機質な機械と向き合って作業をしながら、人の意見を聞くことを必要とし、有機質な作品を作るこの研究は、普通の情報学科ではなく、人文情報学科だからこそふさわしい研究であったと思う。

当初は、何もないところから一つのものを作り出すことは困難を極めたが、自分の力では不足していたところを福田先生にアドバイスを頂きながら作品を作ることで、納得のいくものができたと感じている。また、当初の計画では、実際にクラシック音楽をよく聴く人にも使ってもらい、より質の高いものを作る予定であったが、時間の関係で作品の完成が遅くなってしまったため、福田ゼミの三回生数名にアンケートの協力をしてもらうことしかできなかった。その点では満足はいく作品になったとは言い難い。しかし、自分では見つけられなかった欠点を指摘してもらうことができ、時間の関係上、修正することができずに終わってしまったが、今後の課題が見えてきたので無駄ではなかったと実感している。

この卒業制作と卒業論文を作成することで、音楽データベースの奥の深さを改めて感じることができ、反省点を多く知ることができた。これをふまえて、以後はクラシック音楽だけに限らず、世の中にあるいろいろな CD の詳しいジャンル分けができるようになれば、より『人の役に立つ』ツールになるであろう。

注

- (1) <http://www.justsystem.co.jp/beatjam/>
- (2) <http://www.apple.com/jp/itunes/overview/>
- (3) [http://www.nextftp.com/swlabo/m0\\_pctech/hp\\_classicdb/cdb\\_00\\_to\\_p.htm](http://www.nextftp.com/swlabo/m0_pctech/hp_classicdb/cdb_00_to_p.htm)
- (4) <http://e-words.jp/w/apache.html>

文献表

高橋麻奈

2001 『ここからはじまるデータベース』 日本実業出版社

田中ナルミ/阿部忠光

2003 『標準 MySQL』 ソフトバンクパブリッシング株式会社

紙谷歌寿彦

2003 『はじめての人のためのかんたん PHP + MySQL 入門』 秀和システム