

チベット薬劑オンラインデータベースの 作成について

川邊卓也

目 次

1 はじめに	1
1 テーマを決めた理由	1
2 チベット薬材データベースについて	1
3 データベースの必要性	3
4 制作についての構想	3
5 制作したものについて	4
2 データベースについて	5
1 データベースの制作段階について	5
2 登録されているデータについて	6
3 SQL 上でのデータのリレーションについて	7
3 PHP ファイルの作成について	8
1 一覧表示ページの作成	8
2 詳細ページの作成	11
3 検索機能の追加	13
4 ログイン機能の追加	15
5 編集機能の追加	16
6 画像の編集機能	21
4 おわりに	23
1 作成上での問題点	23
2 自己評価	24
3 今後の課題	25

1 はじめに

(1) テーマを決めた理由

福田ゼミのテーマである「人の役に立つものを作る」という目標に基づき制作した「チベット薬材データベース」は福田先生が所持しているデータを用いてデータベース化する、というものである。このような物を作るようになったきっかけは、先生が「チベットの薬材データベースを作ってみないか？」というお話を聞いたからである。

このデータベースは、私がこれまでに学んできた PHP や MySQL を使うため、自分自身が学んできたものをしっかり理解しているのか。そして、新たに自分で学び作成できるかを確認し、プログラムの動きや意味などを説明できるようにするということ。また、少なからずこのデータベースを利用する人のことを考え、制作することを決めた。

(2) チベット薬材データベースについて

チベット医学は、インドのアーユルヴェーダ医学や中国の漢方医学の影響を受けた伝統医学であるが、現在は国が存在しないことから、チベット医学は中国医学の一部として扱われている傾向にある。

2000年代から（石濱裕美子・谷田伸治・西脇正人・福田洋一）ら4人でこの研究に対し活動しているチベット医学研究会では、チベット医学そのものを位置づけした。

その上で、チベット医学の特徴である多様な薬材に焦点をあてるため、チベットの医学の古典『四部医典』中のは四部医典の第二部「解釈タントラ」の19章～21章にあるチベット医学で用いられる薬材の名前と効能が列記されたところの訳注を作成した。しかし、伝統医学の研究の問題で、『四部医典』において記されている薬材名などは研究者の

調査の違いによりそれぞれ異なった説が提出された。そのため現在のチベット医学の医典では同じ薬材名などれも違いがあり、例えば、1つの薬草に15の学名がついたり、科が違ったりしている。

このデータベースは、『四部医典』に記されている薬剤一つ一つに

1. 【番漢】『番漢薬名 sman ming bod dang rgya'i skad shan sbyar ba bzhugs』『中国本草全書』
2. 【中蔵薬】『中国蔵薬』青海省薬品検験所・青海省蔵医薬研究所主編、上海科学技術出版社、1996年、3巻
3. 【蔵薬巻】『中華本草・蔵薬巻』国家中医薬管理局《中華本草》編委会編、上海科学技術出版社、2002年
4. 【蔵本草】『中華蔵本草』羅達尚編、民族出版社、1997年
5. 【蔵薬志】『蔵薬志』中国科学院西北高原生物研究所・楊永晶主編、青海人民出版社、1991年
6. 【晶鏡】'khrung dpe dri med shel gyi me long (漢名『蔵薬晶鏡本草』)、dga' ba'i rdo rje；西藏昌都地区蔵医院、民族出版社、1995
7. 【青蔵甘南】『青蔵高原甘南蔵薬植物志』、杜品 編著、蘭州：甘肅科学技術出版社、2006
8. 【DTMM】Vaidya Bhagwan *Materia Medica of Tibetan Medicine*. Sri Satguru Publications. Delhi:1994.
9. 【TMP】*Tibetan Medical Paintings*. Ed. Yuri Parfionovitch, Gyurme Dorje and Fernand Meyer, London: Serindia Publications, 1992, 2 vols, 77 plates.
10. 【MTK】*bdud rtsi snying po yan lag brgyad pa gsang ba man ngag gi rgyud las rtsa ba'i rgyud dang bshad pa'i rgyud. The basic Tantra and The Explanatory Tantra from the Secret*

Quintessential Instructions on the Eight Branches of the Ambrosia Essence Tantra. Yuthog Yonten Gonpo, Translated into English by Translation Department, Men-Tsee-Khang, Dharamsala. Dharamsara: Men-tsee-Khang publications, 2008

11. 【BC】*The Quintessence Tantras of Tibetan Medicine*. Tr. Barry Clark. 260 p, Ithaka: Snown Lions Publication, 1995.

以上

11 の現在チベット医学書の文献を用いて合わせることにより。その違いを整理して一覧できるようにし、今後の研究に役立てるために制作された。

(3) データベースの必要性

このデータベースは現代チベット医学書などの作成した本草書類に対応させたチベット医学研究会の研究成果であり、その成果を必要としている研究者に役立てることができる一方で、薬学を研究される人のための資料にもなり、他方面への研究にも役に立つであろうと考える。

私自身は内容ではなく、データベースの機能の追加をすることで研究成果を公開しようとしている研究会にとって役に立つものになると考えている。

(4) 制作についての構想

チベット薬材データベースは、ユーザーモードと管理モードに分かれており、管理モードへのページに行くには、パスワードを入力してログインするという方法をとる。入力してもらったら、管理モードのページ

へ自動で飛ぶように設定する。しかし、ユーザー画面への URL をそのまま入力しても飛ぶことができるので、セキュリティとしての性能は皆無である。

ユーザーモードでは、日本語名、チベット名順にソートをするために、SQL ファイルに専用のテーブルを作成する。新たにテーブルを作成してもデータの量が多いので、データの登録の作業を少しでも楽にするため、半自動登録してくれるプログラムを作る。最初に頂いたデータの登録した順番はチベット名の順番なので、それとは別に日本語での順番のテーブルを作成する。

データの検索機能設置し、必要なデータを取り出せるプログラムを作成する。また、データの一覧表示をするときに、300 以上あるデータを 1 ページでスクロールしつつ見るという状況を作らないために、ページ分割をするプログラムも作成する。

管理モードでは、ユーザーモード時の機能に加え、データや画像の編集ができるようにする。手順は、ページの分割を最優先にして、後に、検索機能を実装し、ユーザーモードの機能を揃えてから管理モードで使用する、編集などをするプログラムを作成していく。

(5) 制作したものについて

- ・ 一度にデータを決めた数だけ取り出すことが出来るページ分割のプログラムを index.php ファイルに追加した。
- ・ 用語の検索をするプログラムを一覧ページに設置して、検索結果の一覧から詳細ページに飛ぶことができるようにした。
- ・ 上記のプログラムを作る上で必要な、検索用に使用するテーブルをデータベースに追加し、検索用語を手動でテーブルに登録するプロ

グラムを作成した。

- ・ 詳細ページからの、次ページ、前ページへのリンクを追加、その上で必要になるデータがアイウエオ順に並んだテーブルをデータベースに追加し、その登録に必要な手動で登録をするプログラムを作成した。
- ・ データベースを編集可能にするため、新たにフォルダを作りユーザー用と管理者用にファイルを差別化させた。管理者用のページからデータの編集をするプログラムについては、データがあるテーブルが2種類あるため、2種類分のプログラムを作成した。
- ・ 画像ファイルの編集するプログラムを作成した。また、実際に設置していないが画像をデータベースから削除するプログラムを作成した。
- ・ 管理者用ページとユーザー用のページを行き来する際に必要なログイン画面を作成した。

2 データベースについて

(1) データベースの制作段階について

チベット薬材データベース自体は、福田先生が制作され私自身は、ある程度作られたものを頂き作成した。であるからして、私が作った物と先生が制作したものを区別しなければならない。

先生が作成されたファイルは以下の物である。

- ・ index.html ・ ・ チベット薬材データベースの top ページ
- ・ index.php ・ ・ チベット薬材名一覧のページ
- ・ detail.php ・ ・ チベット薬材詳細画面のページ

——チベット薬剤オンラインデータベースの作成について——

- ・ footer.php ・ ・ sql を close させたり、Copyright の記述があるファイル
- ・ dbaccess.php ・ ・ データベースに接続するプログラムを書いたファイル、今回は別に connect.php を作成、使用
- ・ sql フォルダ ・ ・ tibmeddb.sql のデータベースなど、薬材の sql データが入っている
- ・ style.css ・ ・ チベット薬材データベースのレイアウトなどが記述されている

以上のファイルが入ったフォルダを最初に頂き、制作を始めた。(1)

また、先生が上記のファイルだけで作成したものが Web で掲載されているのでリンクを書いておく。

http://tibet.que.ne.jp/rgyud_bzhi/index.html

(2) 登録されているデータについて

SQL ファイルに登録されているデータは四部医典のチベット名、訳語、番漢、ロケーションがありそれらは kyou.sql に保存されている。

チベット名はチベット語での名称、訳語は日本語名に直したもの、番漢は現代人が付けた名前ではなく、伝統的な漢訳名でありロケーションは四部医典内での位置である。

詳細の学名にあるデータは、上にある四部医典のデータに関して中蔵薬、蔵薬巻、蔵本草、晶鏡、DTMM、MTK、BC など 11 つの文献で当てはまるデータを表しておりそれぞれに四部医典のチベット名と同じ読み方である異綴や各文献での漢訳と学名、それに対する注記がある。そして、その 4 つを tibmeddb.sql 内の scientific_name テーブル

にまとめている。

画像データは TMPJPG フォルダに入れ、pic.sql にその画像ファイル名のデータなどを保存している。先生に頂いた四部医典や文献データは、各文献の名前で sql ファイルにして、sql フォルダにまとめて保存している。

(3) SQL 上でのデータのリレーションについて

登録されたデータベースの構造や登録した sql ファイルは、福田先生に頂いた物であり、私自身は既存のデータベースに新しいテーブルだけ追加している。

tibmeddb.sql を元にデータベースを作成した、tibmeddb.sql の中身の説明でテーブルは

1. resource テーブル・・・各典抛名のデータがある。
2. kyou テーブル・・・『四部医典』のデータがあり、チベット名と訳語、番館、ロケーションのフィールドがある
3. kkhg テーブル・・・『四部医典』のデータをチベット語の順番にソートしている。
4. scientific_name テーブル・・・現代のチベット医学書のデータが保存されており、異綴、漢訳、学名、注記のフィールドがある。
5. pic テーブル・・・画像ファイルのファイル名のデータがある。
6. kensaku テーブル・・・検索機能を実装した際に作ったテーブルで検索に使うデータがある。
7. order_japanese テーブル・・・kyou テーブルのデータをアイウエオ順に並び替えるために作成した。

以上7つのテーブルの役割を持っている。

各テーブルごとのリレーションは、

- resource テーブル→ resource_id と scientific_name テーブルにある to_resource_id
- kyou テーブル→ kyou_id と kkhg テーブルにある to_kyou_id
他 kensaku, order_japanese テーブル
- kkhg テーブル→ kkhg_id と scientific_name の to_kkhg_id
他 pic テーブル
- scientific_name → to_resource_id と resource テーブルの resource_id 他 kkhg テーブル
- pic テーブル→ to_kkhg_id と kkhg テーブルの kkhg_id
- kensaku テーブル→ to_kyou_id と kyou テーブルの kyou_id
- order_japanese テーブル→ to_kyou_id と kyou テーブルの kyou_id

となっている。(2)

3 PHP ファイルの作成について

(1) 一覧表示ページの作成

(i) require 文で使うファイルについて

PHP のファイルの簡易説明とファイルの動きは、資料でも紹介している。(3)(4)

一覧表示は index.php という名前のファイルで作成しており、そのファイルに入っている機能は、アイウエオ順とチベット語順のソート、用語の検索、四部医典のデータの取出しと表示である。

主となるデータの取出し方法は、まず最初に PHP の require 文を

使って同フォルダにある connect.php を読み込ませる。require 文は別ファイルを読み込ませる機能があり、別ファイルで毎回同じプログラムを使う際に役に立つ。

このファイルで読み込ませた connect.php は、mysql のアクセスとデータベースの選択をするプログラムが書かれており、この操作はデータを取り出すプログラムを使う時には毎回必ずしなければならないのとは他ファイルでも使うという理由で作成した。

また同じ理由で先生が作成した、アクセスした sql を close させるプログラムを書いた「footer.php」も使用している。

(ii) ページ分割

Index.php は kyou テーブルからチベット名、訳語、番館、ロケーションを取り出し表示する。チベット名のところはリンクになっており、クリックするとそのデータの詳細ページに飛ぶように設定してある。その際に kyou_id を送ることによって詳細ページにて学名データを scientific_name テーブルから取り出せるようになっている。

しかし kyou テーブルにあるデータは 300 件以上あり、全部を取り出して表示してもページに収まりきらず、下へ続いてしまい見た目が悪くなる。なんとか見た目を良くしようと考えた結果、データを取り出す件数を 20 件ずつにし、前ページや次ページへと進めるページ分割のプログラムを index.php に追加した。

そのためには MySQL にある Limit 句を使用する。まず初めに、一度に表示するデータを 20 件にする。また、count でデータ数も数えておく。これは PHP の変数宣言で「`lim = 20`」とする。値を 10 にすれば 10 件、40 にすれば 40 件という具合になる次に `$p = intval(@$_GET["p"]);` でデータの位置を取得

する。

これは次ページへのリンクへ飛んだ時などで p の値が変わるが、わかりやすく考えると $p =$ ページ数 (整数) となる。Limit 句は好きなデータの場所から、選択した数だけ表示することができる。

使用例として「*limit5,10*」。この場合 5 番目のデータから 10 件のデータを表示するという具合になる。そこでデータの場所の習得として $st = (p - 1) * lim;$ と変数宣言する。 p の値が 4 なら「 $(4 - 1) * 20 = 60$ 番目から」となる。

最後に「*select * fromkyoulimitst, lim;*」とすることでページ数で取り出すデータの場所から任意の数だけとりだすことができる。これによりデータの取り出す件数も自由に決めることができ、Web ページ内に綺麗に収めることが可能となった。

(iii) アイウエオとチベット語のソートについて

一覧表示のページにチベット語順とアイウエオ順に並び替えるボタンがあり、押したときにページが更新されその順番にソートされるプログラムになっている。

その仕組みは、まずチベット語順にしたければ *kyou* テーブルの *id*、アイウエオ順ならば *kyou* テーブルの *yomi* の順番で並び替えることができる。そのうえでチベット語順にしたければ *name* を *order* に Value を *kyou_id* これを POST で同じページに送る、アイウエオ順ならば *name* を *order* に *value* を *yomi* にする。

送ったデータを *index.php* の一番上で POST でうけとる。その際にデータがなければ *order* の値は *kyou_id* にする。最後にデータを取り出す SQL 文に *order by \$order* にすることで選択した方の順番にソ-

トされる。name を一緒にし、値だけを変えるようにすることにより、無駄なプログラムを書く必要がなくなる。

(2) 詳細ページの作成

(i) 詳細ページについて

詳細ページは detail.php という名前のファイルで作成されている。index.php から送られてきた kyou_id を \$_GET を送り、

```
"select * from kyou where kyou_id=$kyou_id;"
```

で SQL からそのデータ単体を取り出す。また kyou_id を使って

```
"select kkhg_id from kkhg where to_kyou_id=$kyou_id";
```

とし kkhg_id を取り出す。この取り出した kkhg_id は scientific_name テーブルの to_kkhg_id とリレーションしているため、

```
"select * from scientific_name where to_kkhg_id=$kkhg_id"
```

で kyou データ単体に合う異綴、漢名、学名、注記などのデータを取り出すことができる。

しかし kyou データは単体なのに対し、scientific_name にあるデータは複数ある可能性があるので、while で繰り返して取り出す必要がある。また、典拠名だけ resource テーブルの abbr というフィールドがあるので別に取り出す必要がある。これにより、典拠名の異綴、漢名、学名、注記とレコードが完成する。

```
"select abbr from resource where resource_id=$to_resource_id";
```

while の中に上記の SQL 文で典拠名をとりだし最後に print で表示をする。画像ファイルは pic テーブルから取り出すが、pic テーブルにある to_kkhg_id は kkhg テーブルの kkhg_id とリレーションしているので、

```
"select * from pic where to_kkhg_id=$kkhg_id";
```

とすることで、データと一致する画像が取り出すことができる。

(ii) 詳細ページからのページ移動

詳細ページから次ページや前ページに飛ぶ場合、一覧表示のページと同じようにチベット語順とアイウエオ順に分けて制作すると考えると、チベット語順は kyou_id の順番なため、kyou_id の値に 1 を足したり引いたりすることで次ページと前ページへ飛ばすことができるが、アイウエオ順だとそれができない。

そのため yomi の順番にならんだ kyou_id を新しくテーブルを作ることを考えた。そうしてできたのが order_japanese テーブルである。yomi 順のばらばらな kyou_id を同テーブルの to_kyou_id に保存していく、そして order_japanese_id は yomi 順に並んだ kyou_id と同じになる。

データの登録には、自分でボタンを押して登録する手動登録の方法を採用した。もっと効率のよい方法もあると思うが、私自身考え付かなかった。

「zidou_touroku_jp.php」と「zidou_japanese_touroku.php」2つのファイルを作成した。「zidou_touroku_jp.php」ファイルでは、`"select kyou_id from kyou order by yomi;"`で kyou_id を yomi 順に取り出す。とりだした kyou_id を

```
print("<a href=\"zidou_japanese_touroku.php?kyou_id  
=$kyou_id\">$kyou_id\n</a>");
```

このように表示しリンクを送ることで次のファイルへ飛ばす。

2番目の「zidou_japanese_touroku.php」ファイルは、送られてきた kyou_id を \$_GET で受け取り

```
"insert into order_japanese (to_kyou_id) values ('$kyou_id');" ;
```

で kyou_id を order_japanese テーブルの to_kyou_id 順番に登録していく。問題は、一つ一つ順番に押していくので押し間違えると、消すことはできるが最初から登録し直す必要があるということで、今後別の登録方法を考える必要がある。今回は一つ一つボタンを押して登録した。

上記で作成したアイウエオ順のページ移動を使うときは select 文で order_japanese_id を to_kyou_id = kyou_id の条件で取出し、でた order_japanese_id の +1 を次ページ、-1 を前ページにし、それぞれに飛ぶようにリンクをつける。

またページの最初と最後を区別するために、kyou_id の数をカウントし、カウントした数より大きければ、もしくは 1 から始まるので値が 1 より小さければエラー表示することにした。

(3) 検索機能の追加

(i) kensaku.php ファイルの作成

一覧ページから使える検索機能は入力してもらった語句をデータベースの kensaku テーブルにある、search.text に詳細ページのある語句全てを登録するその中から、検索した用語を照らし合わせ表示させる。残念ながら and 検索や or 検索などの機能はない。

kensaku テーブルには id と検索用用語が登録されている search.text そして kyou テーブルの kyou_id とリレーションさせるための、to_kyou_id がある。kensaku.php ファイルでは一覧表示から POST で送られてきたデータを受け取る。送った時の名前は関係ないが、ここ

では sw としている。kensaku.php で

```
"select to_kyou_id from kensaku where search_text like '%$sw%'";
```

という sql 文を使い送られてきた語句が search.text にあるのを条件に、to_kyou_id を取り出す。取り出した to_kyou_id を条件に kyou テーブルからデータを抜き出し、表示する。

あとは一覧ページと同じようにチベット名に詳細ページへのリンクを付ける。

(ii) search.text への検索用語登録

kensaku テーブルの search.text へ検索用語を登録する際に使用したのは detail.php を真似て作った「zidou_kensaku_touroku.php」である。まず detail.php のページに zidou_kensaku_touroku.php へのリンクを付け、その際に kyou_id を送る。

送った kyou_id を条件として kyou テーブルからデータを抜き出す。抜き出したチベット文字、訳語、番館、ロケーションのデータを \$serch_data という変数名に追加していく。

また detail.php と同じように kkhg_id を条件に scientific_name テーブルから、異綴、漢名、学名、注記のデータを抜き出し \$serch_data に追加する。

こうしてまとめたデータを kensaku テーブルに登録する。

```
$sql = "insert into kensaku (to_kyou_id, search_text)
values ('$to_kyou_id', '$search_word')";
```

という SQL 文となる。kyou_id は検索で得た結果のとき、そのデータを kyou テーブルから全て抜き出す SQL 文に使用するため、必ず登録しリレーションしておく。

(iii) 起こった問題と解決方法

上記で述べた方法をもとに検索用語登録ファイルを制作したが、ある文字列がくるときにエラーが起こってしまい、登録できない問題が発生した。その原因は kyou テーブルの番漢フィールドなどにふくまれる「'」という文字である、この記号は kensaku テーブルに insert into でデータを登録するときに、使う '\$search_word' の前後の「'」と同じなため、正常に入力できずエラーとなってしまう。その問題を解決するために、用いたのが PHP の `str_replace` 関数である。

この関数は、特定の文字列を置換し書き換えることができ、使うときは、`str_replace(”置換対象の文字列”, ”置換する文字列”, ”最初の文字列”)` とする。実際に入力するときは

```
$search_word = str_replace( ' , \' , $search_data);
```

となりエラーの原因である「'」を正規表現の\を使い「\'」としてそのままの値で入力するようにしたため。エラーがなくなり、正常にプログラムを動かすことができた。

(4) ログイン機能の追加

(i) 作成の目的

このデータベースには一般の人が閲覧するユーザー用のページと管理者がデータの編集などを行うことができる。管理者用のページと2つのページが差別化されている。そのために、登録されているデータを不正に編集されないために、ユーザー用の一覧ページから管理者用のページへ飛ぶための間にパスワードを入力するログイン画面を作った。

ここで正しいパスワードを入力すると管理者用ページに飛ぶ仕組みとなっている。ただし、セキュリティ要素は全くなく、管理者用ページの

URL を知っていたら、誰でもそのページに飛ぶことができってしまう。パスワードを暗号化させたり、定期的に認証を確認させたり、強制ログアウトさせたりなど、ログイン機能の改良は今後の課題として残っている。

(ii) login.php について

login.php はパスワード入力画面で、正しい値を入力したとき

```
setcookie("TEST", md5(PASSWORD), time() + 3600);
```

で暗号化してクッキーを保存しさらに

```
header("Location: http://....p");
```

とし管理用ページに強制ジャンプするプログラムを設置する。ジャンプした先にも

```
if(isset($_COOKIE["TEST"]) && $_COOKIE["TEST"] != null  
&& md5(PASSWORD) === $_COOKIE["TEST"])
```

という if 文を使いクッキーを確認し、その有無でログインの確認をし、表示するようにする。

間違った値を入力したときは、「パスワードが違います。」というメッセージを表示する。この login.php には index.php の左上から行くことができる。

(5) 編集機能の追加

(i) 詳細ページの編集

データの編集は管理者だけが使える機能でファイルも admin フォルダにいており、他ファイルとは差別化している。そのため index.php と detail.php も hensyu_index.php と hensyu_detail.php にリネームし

てコピーした。

編集が可能な項目は、kyou テーブルにあるチベット名、訳語、番漢、ロケーションと scientific_name テーブルにある異綴、漢訳、学名、注記それぞれを別々に編集できるようにファイルを分けている。

編集ページは入力用ページの「hensyu.php」と入力確認用ページ「hensyu_kakunin.php」最後に更新用ページの「hensyu_koushin.php」の3つのファイルで構成されている。

(ii) kyou テーブルの入力ページ・確認ページについて

hensyu.php ファイルは hensyu_detail.php から送られてきた、kyou_id を用いて sql からデータを取り出し、

```
print("<tr><th>チベット名</th><td><input type='text'  
name='tibetan' size='50' value='$tibetan'</td></tr>");
```

とすることで入力欄を作り、入力した値を次の「hensyu_kakunin.php」ファイルへ POST で送る。

ここで起こる問題は番漢フィールドのデータに「'」あるとき value= の両端にある「'」と同じなので、データが送れないときがある。その場合入力方法を input から<textarea>タグを使って囲み、送ることによってエラーを出さずに送ることができる。

その際の注意点は bankan フィールドの形式が 255 文字までの varchar(255) と設定しているので、設定文字数を超えないようにしなければならない。入力したデータは hensyu_kakunin.php へ送る。

hensyu_kakunin.php ファイルでは hensyu.php ファイルから送られてきたデータを確認することができ、もし間違いがあれば戻るボタンで戻ることができる。確認ページのため編集をできなくし、登録ボタンで次の更新ページへとデータを送る。

(iii) kyou テーブルの更新ページについて

hensyu_koushin.php ファイルは、hensyu_kakunin.php ファイルから送られてきたデータを受け取り編集を実行する。kyou テーブルのデータの編集の場合

```
$sql = "update kyou set tibetan='$tibetan',  
japanese='$japanese', bankan='$bankan',  
location='$location' where kyou_id=$kyou_id";
```

となる。if 文で正常にデータが編集されたかどうかを判断し、表示する。データの編集の一連の流れで kyou テーブルを編集する時は scientific_name テーブルのデータを固定し、scientific_name テーブルのデータを編集するときは、kyou テーブルのデータを編集できないよう固定する。

(iv) scientific_name テーブルの編集

scientific_name テーブルの編集も kyou テーブルの方法と同じように 入力用ページの「hensyu_scientific.php」と入力確認用ページ「hen_kakunin_scientific.php」最後に更新用ページの「hen_koushin_scientific.php」の3つのファイルに分けている。

管理者用の詳細ページである「hensyu_detail.php」から入力ページへいくことができるが、kyou テーブルの編集と違うところは、kyou テーブルのデータは1ページにひとつしかないのに対し、scientific_name テーブルには、1つのページに複数のデータがあるため、kyou テーブルのデータの編集と同じように作成することができない。

そこで先生と相談した結果、scientific_name テーブルから取り出したデータのレコード1行ごとに編集することにきめた。

方法としては `scientific_name` テーブルの `gakumei_id` ひとつが、異綴、漢訳、学名、注記の 1 行のレコードに対応しているため、「`hensyu.php`」から `gakumei_id` を「`hensyu_scientific.php`」に送りそれを条件に取り出すことにした。また送る方法に関しては典拠名をリンク形式にして、編集したいレコードをクリックし、ページ移動とともに `gakumei_id` を送る。

ただし関係のない `kyou` テーブルも見ながら編集できるようにと、`kyou` テーブルからデータを引き出すのに必要な `kyou_id` も同時に送る必要がある。

同じ理由で `resource` テーブルから典拠名を引き出すために必要な `resource_id` もついでに送っておく。最終的に

```
print("<tr><td><a href=\"hensyu_scientific.php?kyou_id=
$kyou_id&gakumei_id=$gakumei_id&resource_id=
$to_resource_id\">$resource_abbr</a>"]
```

とものすごく長いリンクタグになるがエラーが起こることはなかった。

(v) `scientific_name` テーブルの入力ページ・確認ページについて

まず「`hensyu.php`」から送られてきた `kyou_id` と `gakumei_id`、`resource_id` を `GET` で受け取る。受け取った `kyou_id` を使って `kyou` テーブルからデータを取り出し表示する。同様に `resource_id` を使って典拠名を取り出す。

最後に `gakumei_id` を使ってレコード 1 行のデータを取り出すが、並び方は詳細ページのように横ではなく、縦に並べていく。なぜかというところと編集するときには、文字を横に入力するため、できるだけ横にスペースを取っていくためである。

インターネットでよく見るデータベースサイトでも入力欄は横に広

くスペースを取っている。入力欄については、*sql* 登録した際にフィールドで *varchar(255)* など文字数が決まっている。フィールドに関しては *HTML* の<input>タグで *text* など文字数が決まってないフィールドは *HTML* の<textarea>のタグで入力するように合わせていく。上記の *kyou* テーブルでも同じ方法で作る。注意点は、<input>タグの入力欄は高さが固定されていて横に伸びているが、<textarea>タグの入力欄は *box* 形式なので、縦長になってしまう。そこに気を付けてレイアウトする必要がある。

入力したデータは<form>タグで次の「*hen_kakunin_scientific.php*」に送る。「*hen_kakunin_scientific.php*」では、「*hensyu_scientific.php*」から送られたデータを *POST* で受け取り表示する。このページは正しい入力がされているか、確認するだけのページなので、目につく機能はなく、次の「*hen_koushin_scientific.php*」へ送るか、編集ページに戻るかのボタンしかない。ここでも<form>タグを使って更新ページに送る。

(vi) *scientific_name* テーブルの更新ページについて

「*hen_koushin_scientific.php*」は *kyo* テーブル用の更新ページとあまり変わらず。

「*hen_kakunin_scientific.php*」から送られてきたデータを *POST* で受け取り

```
$sql = "update scientific_name set var_name='$var_name',  
chinese_name='$chinese_name', s_name='$s_name',  
notes='$notes' where gakumei_id=$gakumei_id";
```

でデータの更新をする。大事な点はどこを編集するかをはっきりさせるために *gakumei.id* を条件にする必要があるということである。

kyou テーブルの更新ページでも同じだが、最初の「hensyu_detail.php」から3つのファイルを経由する中で、送るのは編集したデータだけでなく id も同様に送らなければならない。これは授業で何回も勉強したが、私自身もたまに忘れてしまうことがあるため、今一度確認をしておく。

最後に if 文を使い編集ができたか、できていないかをか判断させ、編集したあとのデータを<table>タグと echo を使って表示する。

(6) 画像の編集機能

(i) 画像の表示について

画像は詳細ページの学名の下に画像データがあれば貼り付けている。画像自体は php フォルダの TMPJPG フォルダの中に保存されておりデータベース内には pic テーブルに画像ファイルの名前が登録してある。

pic テーブルは to_kkhg_id のフィールドがあり、画像を取り出すときにはその id と kkhg テーブルとリレーションすることで取り出すことができる。取り出したファイル名のデータを \$src という変数に代入し、

```
print "<img src=\"TMPJPG/$src\" />";
```

することで画像の表示をすることができる。また while で取り出しているため、1つのページで複数のファイルを取り出すことで可能である。

(ii) 画像の追加

画像の編集のひとつとして画像の追加を管理者用の「hensyu_detail.php」からできるようにした。作ったファイルは画像ファイル名の入力ページの「gazo_tuika.php」と入力した確認をする「kakunin_gazo_tuika.php」最後に登録する「koshin_gazo_delete.php」

の3ファイルである。

画像の追加は、「追加したい画像を TMPJPG フォルダに入れることができる。」という条件のもとで作成している。具体的な手順はまず「gazo_tuika.php」でフォルダにいった画像のファイル名を入力してもらい、これは、pic テーブルにある画像のファイル名を登録しているフィールドである「jpg_filename」に追加するためである。入力欄は<input>タグで作り、入力してもらったデータは次の「kakunin_gazo_tuika.php」に送られる。

その際に to_kkhg_id も送っておく。「kakunin_gazo_tuika.php」では送られてきたデータを POST で受け取り、ファイル名表示する。これはデータの編集の確認ページと全く同じで、戻るボタンと次のファイルに飛ぶボタンの2つを設置しているだけである。最後の「koshin_gazo_delete.php」は画像の追加を実行し、その際の SQL は

```
$sql = "insert into pic (to_kkhg_id, jpg_filename)
      values ('$to_kkhg_id', '$jpg_filename');";
```

となる。編集ではなく追加になるので update を使わずに insert into の sql 文を使う。

(iii) 画像の削除

画像の追加ができるのなら、削除も作らないといけないと考え作成をした。ただしファイルには問題があり、1 ページに複数の画像がある場合、都合よく画像ファイルを選択して削除するという作業がなかなかうまくいかない。そのため、ファイル自体は作成したが実際に使えるところまでいっていないので、ここでは作ったファイルの説明だけの説明をする。

画像の削除に使用するファイルは、削除する画像の確認をする

「gazo_delete.php」と削除を実行する「koshin_gazo_delete.php」の2つのファイルである。「gazo_delete.php」では「hensyu_detail.php」から送られてきた to_kkhg_id を条件に削除する画像を表示するが、1 ページに複数ある場合、片方だけを選択することができず、そのページすべての画像ファイルが削除対象になってしまう。

ここで削除ボタンを押すと次の「koshin_gazo_delete.php」にデータが送られ。

```
$sql = "delete from pic where to_kkhg_id=$to_kkhg_id;"
```

このような画像を削除する SQL 文が実行される。ここまでが作成したファイルの説明である。今回画像の編集ファイルの作成に取り掛かったのが製作期間の終盤で、他の機能の作成に時間を費やした結果実装を見送ることになった。もう少し工夫をすれば、削除する画像を選択することができるが改良する時間がなく、断念した。

4 おわりに

(1) 作成上での問題点

チベット薬材データベースを作成した中での問題点は、

- ・ 詳細ページから次ページ、前ページへと飛んだ時、稀に kyou_id が登録されていない理由で、データがない部分があり。順番に表示していくことができないこと。
- ・ また次ページ、前ページでデータがない時に、エラーを表示する SQL が完成していないこと。
- ・ 画像ファイルの追加や編集の機能を取り付けたが、1つのページに2つ以上の画像がある場合、画像ファイルを1つ1つ選択して編集できるようにするということができない。これは、画像ファイ

ルを取り出すときに必要な kkhg_id がそのページの画像ファイル全てに対応しているため、都合よく1つのファイルを選択することができないということである。

- ・ scientific_name テーブルの bankan フィールドを編集するとき、に bankan フィールドからデータを取り出した際よくデータに「'」が入っている。これは html の<input>タグの「value='○○'」の両端と同じなためデータを正常に表示することができない。<input>タグを使って登録したかったが、文字の置換をする replace 関数も使うことができなかったため、<text area>タグで登録した。

この3つが大きな問題で、今回の制作では改善することができなかった。

(2) 自己評価

先生と相談してデータベースに取り入れた機能は一通り入れた、今回の制作では検索やページ移動、編集などの機能の追加に集中して作ったため、CSS によるデザインが全くできていなかった。CSS でやったことは、必要最低限の色を付けるのと、データに属性を付けて真ん中や左端に移動させたりと、基本的なレイアウトしかしていない。

検索については、検索した語句に当てはまるもの全てを取り出すだけで、AND 検索や OR 検索にも対応できなかったのが残念である。ページ移動やページ分割は自分が思った通りの結果になり、if 文を使ってエラー処理もするようにしているが、まだ全部確認はできておらず、完璧とは言えない。

これらの機能の追加は授業だけの知識では足りず、参考書で調べたり、インターネットで検索したりと情報を集めた。また、SQL のデータ

内のリレーションについてしっかり理解しておかないと、自分の思った通りのデータの移動や表示ができないと考えた。また、作成の時間配分が上手くいかず、まだまだ作り変えることができる部分が残っている。

(3) 今後の課題

言ってしまうえば実装した機能のほとんどが機能向上させることができる。特にアイウエオ順に kyou_id を並び替える時に作った、手動登録のファイルである。今現在は並び替えて表示した id を毎回自分でクリックして、登録するという方法だが、クリックをしなくても自動で並び替えて、登録をすることができるだろうと私が考える。

今回は制作の都合上作成することができなかったが、今後新しく改変するときには、まずはじめにこの部分を改良していきたいと考える。次に検索機能の追加があり、これは既存の検索機能に AND や OR の検索機能を追加することである。これは、文字の空白なども検索するときの設定に入るので、既存の PHP の知識の他に正規表現まで勉強しなければならない。そのため難易度は高いと考える。

最後に login.php の改良がある。これは、ユーザーと管理のページの差別化を強化する。例えば、管理ページで認証していなければ、パスワードを要求したり、強制的にログアウトさせたり、また履歴を残したり情報の暗号化などができる。その点で考えると、このデータベースでの改良できる部分が一番多いと考える。

また詳細ページにある、次ページ、前ページとページ移動するとき、データがないとエラーを表示するプログラムが実装されていないので改善する必要がある。

注

- (1) 資料 1、初期ファイルを参照
- (2) 資料 2,SQL のリレーションを参照
- (3) ユーザー用で使うファイルは資料 3 を参照
- (4) 管理者用で使うファイルは資料 4 を参照

文献表

石濱裕美子・谷田伸治・西脇正人・福田洋一

2004 「『四部医典』 解釈タントラ 19 章訳注 -チベット医学
の原典解明 (1) -」 『杏雨』 第 7 号

石濱裕美子・谷田伸治・西脇正人・福田洋一

2010 「『四部医典』 解釈タントラ 20 章訳注 -チベット医学
の原典解明 (2)-」 『杏雨』 第 13 号

石濱裕美子・谷田伸治・西脇正人・福田洋一

2012 「『四部医典』 解釈タントラ 21 章訳注 -チベット医学
の原典解明 (3)-」 『杏雨』 第 15 号