створенню відкритих інформаційних систем тощо. Проведено велику роботу по вивченню міжнародного досвіду в цих галузях та виробленню внутрішніх нормативів (положень, інструкцій тощо) в цих ділянках.

Безумовно, стаття не охоплює всі аспекти діяльності нашої бібліотеки, і минулі, і майбутні. Ми запрошуємо колег до співпраці у освоєнні новітніх інформаційних технологій.

УДК 025.326 : 004.628.2

Бочаров Б.П., Воеводина М.Ю., Семененко Л.П.

ГЛОБАЛЬНАЯ КОРРЕКТИРОВКА БД С ИСПОЛЬЗОВАНИЕМ ПРОГРАММЫ AWK

В статье предложена специальная технология, позволяющая значительно облегчить проверку и корректировку больших объемов информации в библиотечном каталоге. Информация экспортируется из электронного каталога в виде iso-файла, затем корректируется с помощью программы awk, переводится в iso-файл и импортируется в электронный каталог. Примеры, описанные в статье, и программу awk можно найти на сайте http://www.lib-journal.ru.

A special technique allowing easy verification and correction of large amount of bibliographic database information is considered in the article. The data are exported from the bibliographic database as an iso-file, corrected with the help of awk-program and then imported back into the bibliographic database. You can find the awk-program and the examples from the article on the site http://www.lib-journal.ru.

Ключевые слова: КАТАЛОГИ ЭЛЕКТРОННЫЕ, БАЗЫ ДАННЫХ, ГЛОБАЛЬНАЯ КОРРЕКТИРОВКА

Эту статью авторы адресуют коллегам, которые уже имеют некоторый опыт в ведении электронных каталогов (или других БД), уже получили от этого изрядную порцию восторгов и разочарований и хотели бы сделать решительный шаг по пути вхождения в глобальные сети, представить там свой информационный продукт как с рекламными, так и с коммерческими целями. Всем тем, кто хочет быть уверен в корректности предоставляемой информации и хочет сохранить и упрочить имидж серьезного партнера. Особенно хочется подчеркнуть, что предлагаемая технология позволит вам ничего не потерять из того, что уже сделано.

Технология, описанная ниже, прошла апробацию в библиотеке Харьковской государственной академии культуры при поиске и исправлении ошибок в БД "Систематическая картотека статей", которая ведется с 1997 г. и содержит свыше 18 тысяч записей.

Почему это для нас важно?

Итак, вы создали свою базу данных и с воодушевлением начинаете ее вести. Работа кропотливая, часто отдает рутиной, а выполнять ее надо быстро. Во-первых, потому, что очень хочется увидеть пользу от ведения БД. А это возможно только при достаточно большом объеме, так как никого (а в особенности вашего начальника) не впечатлит «быстрый поиск» в базе из сотни записей. Во-вторых, почувствовать преимущества ведения именно электронного каталога вы можете только в том случае, когда в нем отражена значительная часть вашего фонда (а в идеале – весь). В такой ситуации ошибки при заполнении самых совершенных входных форм неизбежны! И не надо считать, что они есть «зло», ибо все, что делается людьми, должно быть усовершенствовано ими же. Прибавьте к этому, что вы и ваши ближайшие коллеги осваивают новую область, что одна и та же база данных ведется несколькими сотрудниками параллельно, (а ВУЗы часто еще и студентов привлекают в такой работе в период летней практики), умножьте это на текучесть кадров (ибо квалифицированный сотрудник стоит на рынке труда дороже, а не каждой библиотеке это по карману), учтите, что записи могут создаваться в несколько этапов и пр. Ошибки допускаются разнообразные и неожиданные*:* незаполненные поля, применение недопустимых символов (использование знаков и букв из латиницы и кириллицы, неправильная раскладка клавиатуры), неправильно введенный текст, некорректная разбивка по подполям и т.д.

Когда мы видим ошибки или можем их прогнозировать, то это, прежде всего, показатель высокого уровня профессионализма.

Как быть?

Можно годами просматривать на экране уже созданные записи и так и не «выловить» все ошибки, поскольку наш глаз видит далеко не все, что мы ему поручаем. Хотя этим «прогрессивным» методом поиска ошибок в основном-то все и пользуются. Авторы предлагают принципи-

ально иной подход к этому вопросу и называют его «глобальной корректировкой БД». Такой подход, конечно, требует привлечения к работе программистов.

Под глобальной корректировкой БД мы будем подразумевать корректировку содержимого полей, выполнение которой не связано с редактированием в формах самой БД.

В обычном режиме редактирования в формах самой БД для устранения одной найденной (!) ошибки требуется выполнить ряд операций для изменения содержимого полей. Количество этих операций каждый раз может быть разным и не имеет единого алгоритма. При глобальной корректировке БД вы можете сразу в автоматическом режиме выполнить замену по всей БД.

И еще приятные неожиданности!

Кроме корректировки записей вы попутно можете решить некоторые другие свои насущные проблемы, казалось бы, далекие от проверки правильности заполнения полей БД. Например, вы можете создать свое лингвистическое обеспечение. Да, да, тот самый заветный словарь ключевых слов и предметных рубрик, который вы давно уже готовы сделать (опыт-то есть!), но все руки никак не доходят. Библиотекари-практики прекрасно знают, что даже имея большой опыт, они многократно наступают на одни и те же «грабли»: используют единственное и множественное число в одинаковых понятиях, допускают инверсию, «играются» с приставками, суффиксами и падежными окончаниями. Но это еще полбеды. Многие библиотеки не могут похвастаться профессионалами-предметизаторами, так как утрачены навыки ведения предметных каталогов. Боязнь получить некачественный информационный продукт тормозит процесс импорта – экспорта записей между различными библиотеками. Ликвидировать эту проблему можно либо наличием энциклопедиста-каталогизатора, либо наличием унифицированного лингвистического обеспечения.

Используя описываемую ниже технологию, вы можете унифицировать словари ключевых слов и предметных рубрик и держать их под рукой либо в печатном, либо в электронном виде (как вам больше нравится), можете использовать их для обучения молодых сотрудников - предметизаторов, для информационного обеспечения читателей. *Как создать основу этих словарей, описано ниже.*

Второй пример. Ни для кого не секрет, что стандарты составления библиографического описания меняются, и каждый раз это процесс болезненный: надо решить, что делать с записями, составленными в соответствии со старыми стандартами. Оказывается, используя описываемую технологию, вы можете решить и эту проблему, сэкономив массу времени.

Напрмер, в электронном каталоге книг нашей библиотеки (новые поступления с 1997 г.) есть 11 записей на книги доктора педагогических наук, профессора. Кушнаренко Натальи Николаевны, а в БД «Труды преподавателей» 170 записей на публикации того же автора. Мы хотели бы привести старые записи в соответствие с ныне действующим стандартом (заменить инициалы на имя и отчество, записываемые полностью) и при этом не изменять новые записи, сделанные по стандарту. Если использовать обычную технологию, для корректировки каждой из таких записей нужно: найти каждую запись, просмотреть ее, войти в режим редактирования (**181 раз!**), произвести редактирование (внести в общей сложности **2715 знаков!**), просмотреть запись. При глобальной корректировке после отработки соответствующей программы все записи БД, которые содержат в поле «Автор» запись «Кушнаренко Н.Н.» будут содержать запись «Кушнаренко Наталья Николаевна». При этом и поиск и замена осуществляются автоматически. А список изменяемых терминов может быть любой длины!

Список этот далеко не исчерпан. Авторы приглашают коллег для постановки новых задач, так как возможности технологии достаточно велики.

Алгоритм.

Технология опробована на базе данных «Электронный каталог статей по теме», ведется 4 года, объем – 11 тысяч записей, использована СУБД CDS/ISIS.

Хочется сказать отдельно об особенностях технологии. Программа AWK предназначена для обработки текстовых файлов (советуем ознакомиться со статьей **AWK - универсальная программа работы с текстовыми файлами**/ Бочаров Б.П., Воеводина М.Ю.// Библиотеки учебных заведений, 2002 г., № 4, с.39-53). В связи с этим, работа ведется в несколько этапов по следующему алгоритму.

- Импортируем из БД содержимое записей либо всех, либо для нужного диапазона номеров записей. Получаем ISO – файл (стандарт ISO 2709). Эту операцию выполняет сама СУБД. Полученный файл разбит на строки по 80 символов.

- С помощью программы **del_cr.awk** (см. Приложение) убираем разбиение на сторки. Получаем полноценный ISO – файл.
- С помощью программы **iso2text.awk** (см. Приложение) преобразуем ISO – файл в текстовый (TXT). В этом файле каждая запись будет представлена в таком виде: первая строка-маркер записи; последующие строки – поля, каждое поле на новой строке, каждый экземпляр поля тоже на новой строке, подполя разделяются знаком «^» с последующим кодом подполя. В последней строке приводится номер записи в БД. Именно этот текстовый файл является основным объектом в нашей работе. Именно его мы будем обрабатывать различными программами при глобальной корректировке базы данных.
- Теперь, обрабатывая текстовый файл с помощью соответствующих программ, можно выполнить намеченное. Если вы занимаетесь только поиском ошибок, то получаем файл с перечнем ошибочных записей и исправляете их вручную. Если вы выполняете преобразование текстового файла, переходим к следующему пункту.
- Когда мы сделали со своим текстовым файлом все, что хотели, надо вставить его на место. Для этого из текстового он должен снова стать ISO – файлом. Выполняем это с помощью программы **text2iso.awk** (см. Приложение).
- И последнее. Выполняем импорт нашего улучшенного корректировкой ISO – файла БД. Напоминаем, что импорт надо производить заменой, а не добавлением! Эту операцию выполняет сама СУБД.

С чего начать?

Предлагаем с простого. В каждой БД есть контролируемые поля, которые всегда должны быть непустыми. Например, для нашей БД это поле «Источник», поле «Каталогизатор» и др. На первом этапе работы хотелось бы знать, в каких записях эти поля не заполнены. Для этого используем программу **pust_015.awk**, которая из полученного текстового файла «вытащит» для нас просто номера записей, которые надо исправить. Тест программы приводится в каталоге **part0** приложения. В тексте вы можете видеть регулярное выражение /^015/, это соответствует полю «Источник» для нашей базы данных. На выходе вы получаете текстовый файл, в котором перечислены номера записей, в которых отсутствует поле 015. Остается только внести исправления, зная номера записей это можно сделать вручную.

Поменяв регулярное выражение на /^082/ вы получите аналогичный список для поля «Каталогизатор». И так можете сделать проверку для всех интересующих вас полей. Трудозатраты на такую операцию не сравнимы с трудозатратами на прямой просмотр!

Второй по сложности этап – это нахождение в полях и подполях недопустимых символов. Например, в поле «Индекс ББК» не может быть знака «+», наличие же его в этом поле говорит о том, что поле заполнено неправильно и поиск будет вестись некорректно. Программа **bbk_pl.awk**, текст которой приведен в каталоге **part0** приложения сформирует для вас текстовый файл, в котором перечислены номера неправильно заполненных записей. Аналогично вы можете обрабатывать другие поля, делая замены в соответствующих регулярных выражениях.

Ну, и наконец-то самое интересное. Самый сложный и самый впечатляющий этап вашей работы - это автоматическая вставка скорректированного поля или подполя. С помощью программы **get_fld.awk** (см. Приложение) получаем текстовый файл с содержимым поля (или подполя). Какого? В нашем примере это поле 078 «Ключевые слова» и подполе 211^c «Инициалы автора». Это также может быть поле 005 «Предметные рубрики», 009 «Индекс ББК» или какое-либо другое. Теперь нужно произвести корректировку. Это можно сделать несколькими способами. Авторы считают удобным воспользоваться таблицей WORD. В одну колонку таблицы вставляем содержимое извлеченных полей. Во второй колонке должны появиться соответствующие новые значения полей. Естественно, то, что не подлежит корректировке, просто копируем. Таким образом, каждому извлеченному из БД значению ставим в соответствие исправленное либо прежнее значение. Таблицу преобразовываем в текстовый файл. Все готово для автоматической вставки новых значений поля. Осуществляется она программой **rpl_fld.awk** (см. Приложение).

На основе исправленного списка содержимого поля «Ключевые слова» можно начинать формировать основу своего лингвистического обеспечения. Впрочем, это уже совсем иная задача, которая остается за пределами нашего повествования.

Приложение

На сайте журнала **www.lib-journal.ru** находятся все программы, реализующие предложенную технологию. Ниже подробно описан алгоритм их работы.

Шаг 1. Перевод файла, экспортированного из ISIS в текстовый вид.

В каталоге **part1** приложения находятся следующие файлы:

src.iso - файл, экспортированный из ISIS,
Del_cr.awk - программа, удаления лишних концов строк,
iso2text.awk - перевод iso-файлов в текстовый вид,
i2t.bat - вызов программ **del_cr.awk** и **iso2text.awk**.

Рассмотрим подробно, как работают эти программы. Для выполнения операции перевода файла, экспортированного из ISIS в текстовый вид необходимо выполнить следующий командный файл: **i2t.bat**

Сначала выполняется программа, исправляющая изуродованный ISIS'ом iso-файл.

В результате работы этой программы появляется файл промежуточный файл **cor.iso**.

Затем вызывается программа, перевода iso-файла в текстовый вид.

Напомним, что запись в формате ISO 2709 начинается с маркера (24 символа). Затем идет справочник, который состоит из нескольких статей, длиной 12 символов (по количеству полей в записи). Первые три символа в статье – номер поля. В конце справочника ставится символ - разделитель полей. Дальше записывается содержание полей, разделенных на подполя. После каждого поля ставится разделитель полей. В конце записи ставится символ – разделитель записей.

Мы собираемся записывать измененную информацию в iso-файл, поэтому нужно сохранить в маркере все поля, не относящиеся к длине записи и адресам данных внутри записи. В маркере первые 5 символов - длина записи, а 5 символов, начиная с 13-го – базовый адрес данных (смещение 1-го подполя).Эти действия выполняются программой **iso2text.awk**

В результате работы программы появляется текстовый файл **cor.txt**.

Шаг 2. Выбор и анализ содержимого конкретного поля.

В каталоге **part2** приложения находятся следующие файлы:

cor.txt - файл, полученный на предыдущем шаге,
get_fld.awk - программа, выбирающая из текстового файла значения конкретных полей (подполей),
go.bat - вызов программ **get_fld.awk**.

Мы будем использовать программу, позволяющую вывести значение любого поля (подполя). Вызов этой программы осуществляется с помощью командного файла **go.bat** следующим образом:

go XXXa,
XXX – номер поля (ведущие нули обязательны, например 005),
a – буква подполя (может быть опущена).
Имя выходного файла: **XXXa.val**.
Примеры вызова:

Вызов	**Выполняемое действие**	**Выходной файл**
go 211a	Выбор фамилий авторов	**211a.val**
go 078	Выбор ключевых слов	**078.val**

Сначала мы записываем значения полей в промежуточный файл **XXXa.zzz**, затем сортируем этот файл. Результат сортировки – файл **XXXa.val**. В конце работы мы удаляем промежуточный файл.

Комментарии в тексте программы **get_fld.awk** подробно описывают ее работу.

Применим эту программу для исправления ошибок в исходном файле.

Сначала проверим поле **211c** – инициалы автора. С помощью команды **go 211c** выведем значения этого поля в файл **211c.val**. Заметим, что в программе **get_fld.awk** мы два раза выводим одни и те же значения. Это сделано для удобства проверки информации в Word.

Переводим значения поля **211c** в таблицу Word:

1. В текстовом редакторе FAR копируем содержимое файла **211c.val** в буфер обмена.
2. Вызываем Word и создаем файл **211c.doc**.
3. Вставляем в этот файл информацию из буфера обмена.
4. Выделяем всю вставленную информацию и вызываем пункты меню «Таблица – Преобразовать – Текст в таблицу». В появившемся окне указываем «количество колонок» - 2.

Информацию в левом столбце оставляем без изменений, а в правом столбце исправляем ошибки. В данном случае в первой строке таблицы (ошибочное значение подчеркнуто) добавляем пропущенное отчество в инициалах автора. У нас получилась такая таблица:

Я.	Я.Л.
Я.Л.	Я.Л.

Заметим, что программы FAR и Word автоматически перекодировали текст из кодировки DOS в Windows.

Для дальнейшего изменения информации в iso-файле нам необходимо перевести эту таблицу в текстовый вид и произвести обратную перекодировку из Windows в DOS. Алгоритм действий такой:

1. Сохраняем файл **211c.doc** в текстовом виде (меню «Файл – Сохранить как», тип сохраняемого документа «только текст»).

2. Закрываем Word и вызываем на редактирование в FAR файл **211c.txt** (только что полученный из Word). Копируем содержимое файла в буфер обмена. Выходим из редактора.

3. Создаем новый файл **211c.rpl**. Устанавливаем его кодировку как DOS. Вставляем информацию из буфера обмена.

4. Удаляем все пустые строки в конце файла.

Файл **211c.rpl** будет использован на этапе замены информации в текстовом файле.

Аналогичным образом получаем файл **078.rpl** – файл ключевых слов (все подполя поля **078**).

Сразу видно, что в названии CDS/ISIS хотя бы один раз была допущена ошибка (первая буква не латинская, а русская). Фрагмент файла **078.doc** представлен ниже, ошибочное значение подчеркнуто.

программное обеспечение	программное обеспечение
CDS/ISIS	CDS/ISIS
системы автоматизированные информационные	системы автоматизированные информационные

Шаг 3. Замена в текстовом файле.

В каталоге **part3** приложения находятся следующие файлы:

cor.txt - файл, полученный на шаге 1,
078.rpl - файл замен для поля **078**,
211c.rpl - файл замен для подполя **211c**,
rpl_fld.awk - программа, заменяющая значения полей (подполей) в текстовом файле,
go.bat - командный файл, осуществляющий последовательную корректировку подполя **211c** и поля **078**.

Вызов программы **rpl_fld.awk** аналогичен вызову программы **get_fld.awk**, описанной выше.

Имя входного файла замен **XXXa.rpl** (см. шаг 2).

Если значение подполя не найдено в файле замен, то информация об этом подполе выводится в файл ошибок **XXXa.err**.

Командный файл **go.bat** последовательно вызывает программу **rpl_fld.awk** для замены значений подполя **211c** и поля **078**. После изменения поля **211c** информация записывается в файл **tmp.zzz**, который является входным для замены поля **078**. Затем этот временный файл удаляется.

В результате получается файл **new.txt**.

В результате получается файл **new.txt**.

Шаг 4. Преобразование текстовых файлов в iso-файлы.

В каталоге **part4** приложения находятся следующие файлы:

new.txt - текстовый файл, который нужно преобразовать, в iso-файл.
text2iso.awk - преобразование текстовых файлов в iso-файлы,
t2i.bat - вызов программы **text2iso.awk**.

При преобразовании текстового файла в iso-файл нам нужно решить следующие задачи:

1. Записать в справочник номера полей, содержание полей сохранить отдельно.
2. Определить и записать числовую информацию в каждую статью справочника. За номером поля (3 символа) должны следовать длина содержания поля (4 символа) и позиция начального символа (5 символов).
3. Заполнить в маркере длину записи (первые 5 символов) и базовый адрес данных (5 символов, начиная с 13-го).
4. Записать в файл маркер, справочник и содержание полей.

Все эти задачи решает программа **text2iso.awk.**