УДК 519.6

БАЗА ДАННЫХ СИСТЕМЫ “АРЕОЛА”

П. А. Брызгалов[1]

В статье описывается структура базы данных системы “Ареола” — программной оболочки для создания электронных энциклопедий, начальное занесение данных, поиск ошибок и редактирование базы данных. Работа поддержана грантом РФФИ № 03–07–90427в.

Ключевые слова: электронные справочники, электронные энциклопедии, мультимедия, база данных, MySQL, Java.

1. Введение. Бурное развитие Интернет-технологий и различных видов дистанционного образования подтолкнуло нас к идее создания электронной версии справочника по линейной алгебре. В результате была создана система “Линеал”, доступная для работы через Интернет по адресу lineal.guru.ru. Она была подробно описана в [1]. Техническая основа, на которой построена система “Линеал”, оказалась пригодной для создания новых справочников и энциклопедий по совершенно различным предметным областям, и сейчас начата работа над энциклопедией по параллельным вычислениям. Эта техническая основа получила название “Ареола”. В этой статье мы подробно рассмотрим базу данных, в которой система хранит всю информацию о предметной области.

2. Структура базы данных. Рассматриваемая система “Ареола” позволяет создавать электронные энциклопедии по различным областям знаний. Как и в любой энциклопедии, информация представляется в виде отдельных статей, но в отличие от традиционных энциклопедий система “Ареола” позволяет присваивать статьям различные атрибуты, что значительно расширяет возможности пользователя по поиску нужной информации. Подробно устройство и работа системы были рассмотрены в [2]. Вся информация о предметной области хранится в базе данных, и в настоящей статье мы рассмотрим только саму базу данных, ее начальное наполнение информацией, поиск и исправление ошибок и те служебные программы, которые помогают это делать.

Как упоминалось выше, вся информация о предметной области разбивается на статьи. Каждая статья — это отдельная запись, строка в главной таблице базы данных. Поля этой таблицы — атрибуты статей — подробно рассматриваются в следующем пункте. Статьи организуются иерархически по разделам, главам и параграфам. Кроме того, сами разделы, главы и параграфы также хранятся в базе данных в виде отдельных записей. Каждая статья имеет свой номер. Статьи различаются по уровню сложности и по типу. Между статьями устанавливаются направленные логические отношения. Например, это может быть порядок изучения статей, т.е. если в одной статье используются факты или термины, описанные в другой, то первая является логическим следствием второй. Именно наличие логических отношений существенно облегчает поиск нужной информации тем пользователям, которые хотят узнать не отдельные факты, а получить систематизированные знания по предметной области.

3. Перечень таблиц и полей и их назначение. Основная информация о статьях собрана в главной таблице базы данных, в которой записи соответствуют отдельным статьям и разделам предметной области.

Главная таблица содержит следующую группу полей, относящихся к номерам статей.

Номер раздела — номер того раздела, в который входит статья или подраздел. Устанавливается ненулевое значение во всех строках таблицы. Для разделов это — номер самого раздела.

Номер главы — номер той главы, в которую входит статья или подраздел. Для разделов он не определен.

Номер параграфа — номер того параграфа, в который входит статья. Для параграфов — номер самого параграфа внутри главы. Для разделов и глав не определен.

Номер статьи — номер статьи внутри главы. Хотя статьи группируются по параграфам, номер параграфа не входит в номер статьи. Это означает, что статьи имеют сквозную нумерацию внутри главы.

Номер-строка — комплексное строчное представление номера подраздела или статьи. Для разделов он состоит из одного числа — номера самого раздела. Для глав — это номер раздела, точка, номер главы

[1] Научно-исследовательский вычислительный центр, Московский государственный университет им. М. В. Ломоносова, 119992, Москва; e-mail: pyotr777@guru.ru

в разделе. Для параграфов — это три числа, разделенные точками: номер раздела, номер главы и номер параграфа внутри главы. Для статей — это тоже три числа, только первые два — номер раздела и номер главы — разделены точкой, а перед последним — номером статьи в главе — стоит дефис.

При начальном заполнении базы данных указанные номера берутся из исходного текстового файла, который подготавливается заранее, где они указаны в виде номеров-строк. При этом заполняются поля "номер раздела", "номер главы", "номер статьи" и "номер-строка". Информация о номерах и названиях параграфов также берется из текстового файла, но ее формат отличается от формата информации о статьях, разделах и главах. Формат информации о параграфах таков: после специальной метки начала информации о параграфе идет номер-строка параграфа, его название и список номеров-строк статей, входящих в этот параграф. Поля "номер раздела" и "номер главы" для параграфов заполняются аналогично случаю со статьями — они берутся из номеров-строк.

Кроме того, главная таблица содержит следующие поля.

Название — название подраздела или статьи.

Текст — текст статьи. Для подразделов поле не определено.

Дополнение — дополнительная текстовая информация. Это могут быть комментарии к статьям, доказательства, примеры и т.д. Для подразделов это поле не определено.

Эти три поля заполняются на основе информации из текстового файла. Вся текстовая информация представляется в формате HTML. В текст можно вставлять рисунки. Это могут быть пояснительные иллюстрации, фотографии, специальные символы, математические формулы и т.д. Более того, можно даже использовать мультимедийные вставки и многое другое: все виды информации, доступные для представления в Интернете. Все это позволяет делать язык разметки HTML. Непосредственно в базе данных сохраняется текст со ссылками на объекты, а сами объекты хранятся в виде файлов на сервере.

Тип — тип статьи: определение, дополнение, утверждение и т.п. Для подразделов это поле также не определено.

Уровень сложности — уровень сложности статьи. Для подразделов это поле не определено. Хотя при работе с графами логических отношений можно отображать уровень сложности подразделов, там он вычисляется во время работы и определяется как высший уровень сложности входящих в подраздел статей, но только тех, что присутствуют в выборке, по которой строится граф. Таким образом, отображаемый в графе уровень сложности подразделов будет зависеть от набора статей, по которым строится граф.

Номера предшественников — список номеров-строк логических предшественников через запятую. Это поле используется в основном для удобства создателей системы, а при работе системы информация о предшественниках берется из другой таблицы, состоящей из двух полей: номер-строка предшественника — номер-строка следствия. Такую таблицу проще использовать для поиска предшественников и следствий и работа с ней происходит быстрее.

Значения полей "тип", "уровень сложности" и "предшественники" подставляются из текстового файла. Таблица пар "предшественник-следствие" заполняется автоматически на основе этой информации.

Индекс — число, вычисляемое на основе иерархического положения статей и подразделов, которое служит для упорядочивания записей из таблицы. Вычисляется автоматически.

Количество предшественников и следствий — это восемь полей: по количеству уровней сложности (четыре) для предшественников и для следствий. Не все поля бывают заполнены. Если уровень сложности статьи выше минимального, то поля, соответствующие предшественникам и следствиям с меньшим уровнем сложности, не заполняются, т. к. в этом случае сама статья никогда не будет использоваться.

Значения указанных полей рассчитываются автоматически.

На основе информации о подразделах из главной таблицы формируется структурный указатель (оглавление). Кроме этого, в базе данных в отдельной таблице хранится предметный указатель — список терминов и соответствующих им номеров статей. Информация для предметного указателя берется из текстового файла.

4. Подготовка текстовых файлов. Итак, для заполнения системы информацией о предметной области необходимо подготовить текстовые файлы следующих трех видов:

— информация о разделах, главах, параграфах и статьях,

— информация о номерах статей, входящих в параграфы,

— информация о предметном указателе.

Рассмотрим подробнее каждый из трех видов.

Информация о разделах, главах, параграфах и статьях включает в себя: номера-строки всех разделов, глав, параграфов, статей и их названия. Кроме того, только для статей дополнительно необходимы: текст и дополнение, тип, уровень сложности и список номеров-строк предшественников. Формат информации

таков: после меток с названиями полей через пробел или с новой строки идут содержания соответствующих полей, которые заканчиваются либо перед следующей меткой, либо концом файла.

Информация о номерах статей, входящих в параграфы, задается в похожем формате: после меток идут номера параграфов и списки входящих в указанные параграфы статей через запятую.

Информация о предметном указателе задается в следующем виде: номер-строка статьи, разделитель, термин, определяемый в указанной статье. Кроме того, для удобства навигации по указателю перед терминами, идущими первыми на каждую из букв, ставятся HTML-метки, причем перед кавычками ставится символ "\" (например: <a name=\"b\"> для буквы "б"). Названия меток — представление русских букв в системе транслитерации.

5. Программа для загрузки информации в базу данных. После того как описанные выше текстовые файлы со всей необходимой информацией готовы, можно приступать к загрузке информации в базу данных. Для этого служит специальная программа на языке Java, входящая в состав системы.

Программа не создает базу данных и входящие в нее таблицы, это нужно сделать самостоятельно. Конкретная последовательность действий при создании базы данных зависит от используемой СУБД. Программа подключается к базе данных через мост JDBC : ODBC, поэтому после создания базы нужно еще создать источник данных ODBC. В системе Windows это делается через Контрольную панель, Администрирование, Источники данных (ODBC). После этого запускается программа для загрузки данных.

В программе подключение к базе данных реализуется следующим образом. Создается переменная-строка, в которую записывается имя базы данных:

```
url = ‘‘jdbc:odbc:’’+BDname;
```

где BDname — имя ODBC источника этой базы. Затем устанавливаем соединение с базой данных:

```
try {
      Class.forName(‘‘sun.jdbc.odbc.JdbcOdbcDriver’’);
    } catch (ClassNotFoundException e) {
      System.out.println(‘‘Class not found’’);
    }
try {
      con = DriverManager.getConnection(url,’’имя’’,’’пароль’’);
    } catch (SQLException ex) {
      System.out.println(‘‘Driver manager exception. State = ‘‘+ex.getSQLState()+’’\n’’)
      System.out.println(‘‘msg = ’’+ex.getMessage());
    }
```

где con — объект класса Connection. Если для доступа к базе данных не требуются имя пользователя и пароль, они не указываются в команде getConnection.

Запись данных выполняется при помощи переменной класса Statement:

```
Statement stmt;
try {
      stmt = con.createStatement();
    } catch (SQLException ex) {
      System.out.println(‘‘Error creating statement in saveData’’);
    }
```

Подготавливается SQL-выражение для записи данных, которое сохраняется в переменной класса String sql, затем выполняется непосредственно запись в базу данных:

```
try {
      int ins = stmt.executeUpdate(sql);
    } catch (SQLException ex) {
      System.out.println(‘‘SQL exception. SQL=’’+sql+ ‘‘\n’’+‘‘state = ’’+ex.getSQLState()+
                         +‘‘\n’’)
      System.out.println(‘‘code = ’’+ex.getErrorCode());
    }
```

Программа загружает информацию из одного файла за один раз. Рекомендуемая очередность такова: сначала загрузить информацию из файла с номерами, названиями, текстами и другой информацией о разделах, главах, параграфах и статьях. Перед загрузкой информации из этого файла необходимо обратить

внимание на формат поля “уровень сложности”. Он может быть представлен в виде числа от единицы до четырех, либо оценками: 3, 4, 5 и 5+. Для переключения форматов служит поле “корр.у.с.” в программе. Если уровень сложности представлен в виде оценок, то надо поставить галочку.

Затем следует загрузить информацию из файла с номерами статей, входящих в параграфы. Последним рекомендуем обработать файл с информацией о предметном указателе. Поскольку формат данных для предметного указателя прост, программа не предназначена для работы с ним, а данные оформляются в виде SQL-команд и заносятся стандартными средствами СУБД.

После загрузки информации из файлов необходимо еще сгенерировать служебную информацию, необходимую для работы системы. Это — структурный указатель и количество предшественников и следствий, а также значения поля Nindex, служащего для упорядочивания записей о разделах, главах, параграфах и статьях.

Программа автоматически формирует структурные указатели для каждого уровня сложности при нажатии на соответствующую кнопку. В программе формируется лишь тело указателя, т.е. та часть, которая зависит от конкретной информации о предметной области, а неизменяемая часть подставляется из двух специальных файлов. Готовые файлы со структурными указателями потом необходимо поместить в директорию на сервере, где хранятся все файлы системы с расширениями html и php.

Количества предшественников и следствий рассчитываются в программе и подставляются в соответствующие поля главной таблицы базы данных при нажатии на соответствующую кнопку.

6. Программа для поиска ошибок. Одной из главных задач системы является отображение внутренней логической структуры предметной области, представленной в виде направленных логических связей между статьями, на ориентированный граф. На эти связи накладываются ограничения. Во-первых, нельзя, чтобы статья с меньшим номером была логическим следствием статьи с большим номером. Это ограничение введено для того, чтобы в графе логических отношений не было контуров. Во-вторых, нельзя, чтобы статья с меньшим уровнем сложности была логическим следствием статьи с большим номером. Если возникнет такая ситуация, соответствующая логическая связь может быть прервана при ограничении уровня сложности.

Для нахождения описанных ситуаций служит другая специальная программа. Данная программа также позволяет редактировать основные поля любой записи основной таблицы базы данных. Эта программа написана на языке Java и подключается к базе данных через ODBC.

7. Внесение дополнений и исправлений. Упомянутые выше две программы могут быть использованы для внесения дополнений и исправлений. Дополнения — новые статьи — вносятся в базу данных при помощи той же программы, которая используется для начальной загрузки данных. Порядок внесения дополнений ничем не отличается от порядка начальной загрузки и состоит из трех этапов: загрузки информации о статьях из текстовых файлов, генерация структурного указателя и расчет количества предшественников и следствий. Новые файлы для структурного указателя необходимо положить на место старых. Отдельно стоит упомянуть о картинках и других мультимедийных вставках, которые могут присутствовать в новых статьях. Необходимо, чтобы имена новых файлов не пересекались с именами уже существующих. Если проверить уникальность имен сложно, рекомендуется складывать дополнительные файлы — картинки и другие мультимедийные вставки — в новую директорию, а в текстах новых статей указать путь к вновь созданной директории во всех ссылках на мультимедийные файлы.

Исправления в базу данных можно вносить при помощи той же программы, которая служит для обнаружения логических ошибок. Эта программа позволяет вывести на экран информацию о названии, тексте, дополнении, типе, уровне сложности и предшественниках любой статьи. Для этого надо указать имя ODBC-источника базы данных, номер и название того поля, которое интересует. Выведенную на экран информацию можно редактировать в окне программы, или, скопировав его содержимое, в любом текстовом редакторе. После правки обновленную информацию можно записать в базу данных, при этом старая информация затирается.

СПИСОК ЛИТЕРАТУРЫ

1. *Воеводин В.В., Воеводин Вл.В.* ЛИНЕАЛ: электронная энциклопедия по линейной алгебре // Вычислительные методы и программирование. 2002. **3**, № 1. 131–140.
2. *Брызгалов П.А.* Система “Ареола” — программная оболочка для создания электронных энциклопедий // Вычислительные методы и программирование. 2005. **6**, № 1. 136–140.

Поступила в редакцию
28.04.2005