



УДК 004.657

АНАЛИЗ МЕТОДОВ ОБРАЩЕНИЯ К БАЗЕ ДАННЫХ ИЗ WEB-ПРИЛОЖЕНИЯ*Л.И. Саетова, saetova_stud@mail.ru*

Нижекамский химико-технологический институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Казанский национальный исследовательский технологический университет»

В данной статье рассматривается разработка механизма обращения к имеющейся базе данных FoxPro из web-приложения. В статье проводится анализ возможных способов решения проблемы. В результате исследования разработан механизм обращения к БД и проведена его апробация с использованием локального сервера OpenServer.

Ключевые слова: база данных, веб-приложение, dbf-файл, dbf-таблица.

RESEARCH THE METHODS OF ACCESSING TO THE DATABASE FROM WEB-APPLICATION*L.I. Saetova*

Nizhnekamsk Institute of Chemical Technology (branch of) Federal State Budget Institution of Higher Education «Kazan National Research University of Technology»

The article under review reveals the development of mechanism for accessing the existing FoxPro database from a web-application. The article contains the technical task for the development of a web-application and also the analysis of possible ways of solving the problem is carried out. As a result of the research, a mechanism for accessing the database was developed and its validation was carried out using the local server OpenServer.

Key words: data base, web-application, dbf-file, dbf-table.

Введение

Бухгалтерский учет Нижекамского химико-технологического института ведется в системе автоматизации «Парус-Бюджет 7», представляющей собой программный продукт, построенный на базе архитектуры файл-сервер с использованием СУБД FoxPro. Под учет оплаты за обучение студентами адаптирован один из модулей системы «Склад». В нем ведется список обучающихся студентов, их лицевых счетов, выставляемых им исходящих счетов на оплату и журнал учета оплаты за обучение на основе реестров, предоставляемых банком. Такой способ автоматизации работы отдела по работе со студентами, обучающимися на внебюджетной основе, имеет как свои достоинства, так и недостатки. Несомненным плюсом является то, что большая часть функций отдела автоматизирована, есть наработанная более чем за 10 лет база данных со списком всех когда-либо обучавшихся за этот период студентов, их платежей и выставляемых им счетов на оплату. Недостаток такого способа автоматизации заключается в том, что модуль все же не специализирован под конкретные задачи отдела, а лишь адаптирован под выполнение некоторых из его функций.

ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ

С точки зрения хранения информации и доступа к ней в веб-приложении важным элементом является наличие сервера базы данных. Обычно данные хранятся в специальном SQL-сервере. Он обслуживает базу данных и отвечает за целостность и сохранность данных, а также обеспечивает операции ввода-вывода при доступе клиента к информации. Специфика сервера базы данных заключается



в том, что данные, как правило, обрабатываются транзакционно, т.е. система запрашивает небольшой объем данных, проводит над ними операцию и затем сохраняет. Таким образом, веб-приложение может работать по двухзвенной архитектуре клиент-сервер (рис. 1). Т.е. распределение трех базовых компонентов (компонент представления данных, прикладной компонент и компонент управления ресурсом) происходит между двумя узлами (клиентом и сервером). Здесь сервер напрямую отвечает на клиентские запросы, используя только собственные ресурсы. Т.е. сервер не вызывает сторонние сетевые приложения и не обращается к сторонним ресурсам для выполнения какой-либо части запроса [1].

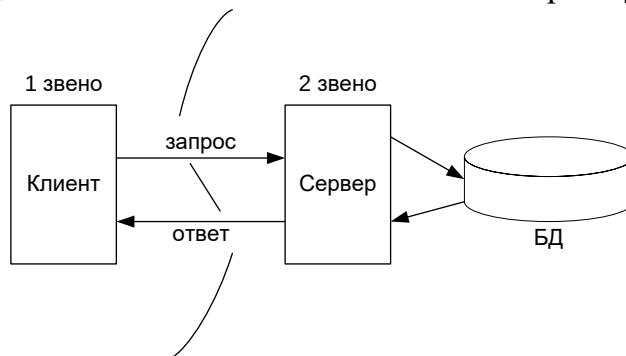


Рис. 1 – Двухзвенная архитектура клиент-сервер

Настольные же приложения используют файловую систему хранения данных. Они работают по файл-серверной технологии. Файловый сервер предоставляет доступ к файловым ресурсам, в том числе и к удаленной базе данных. В этом случае выделенный узел сети является файловым сервером, на котором размещены файлы базы данных. На клиентах выполняются приложения, в которых совмещены компонент представления и прикладной компонент (СУБД и прикладная программа), использующие подключенную удаленную базу как локальный файл. Протоколы обмена при этом представляют набор низкоуровневых вызовов операций файловой системы.

Такая модель неэффективна ввиду того, что при активной работе с таблицами базы данных происходит большая нагрузка на сеть. Частичным решением является поддержка тиражирования (репликации) таблиц и запросов. В этом случае, например, при изменении данных, обновляется не вся таблица, а только модифицированная ее часть.

В случае ведения учета денежных поступлений на оплату обучения в институте реализована файл-серверная технология, где один узел сети выделен под размещение файлов базы данных, а на клиенте выполняется настольное приложение, в котором ведется учет (с использованием СУБД FoxPro) (рис. 2).

Как уже говорилось ранее, такое решение имеет свои недостатки. Поэтому в дополнение к существующему настольному приложению было решено разработать веб-приложение, позволяющее просматривать актуальные данные по учету внебюджетных средств, представленные в виде различных отчетов. Однако реализация веб-приложения подразумевает его интеграцию с существующей базой данных, представленной набором таблиц формата dbf.

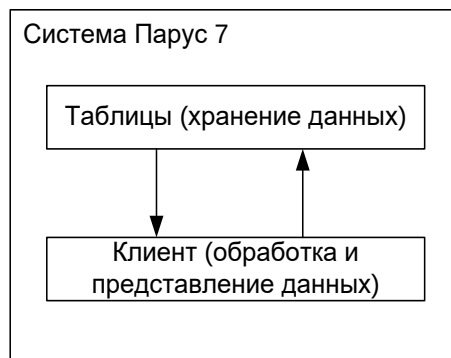


Рис. 2 – Архитектура файл-сервер

Таким образом, *целью* данного исследования является определение метода обращения к данным существующей базы данных из разрабатываемого веб-приложения.

ОПИСАНИЕ СУЩЕСТВУЮЩИХ СПОСОБОВ ДОСТУПА К БД

База данных системы Парус 7 Бюджет, используемая для ведения учета, представляет собой множество dbf-файлов, в которых хранится вся информация.

Все файлы формата dbf относятся к формату хранения данных, который используется в роли одного из нескольких стандартных методов хранения и передачи информации посредством систем управления базами данных, электронных таблиц и так далее.

Однако dbf-файл не имеет официально утвержденной стандартизации, поэтому весьма сложно утверждать возможность какой-либо произвольной утилиты открыть файл, формат хранения которого содержит расширение dbf. В содержание файла dbf входит заголовок, в котором хранится информация о структуре базы данных и количестве допустимых записей, а также область данных, состоящая из последовательно организованной таблицы с записями фиксированной длины.

Таким образом, необходимо определить способ предоставления доступа веб-приложения к информации, хранящейся в dbf-таблицах. Рассмотрим некоторые из них.

1. Преобразование dbf-таблиц в CSV-файлы и последующий их импорт в MS SQL Server. CSV (Comma-Separated Values) – это простой текстовый формат, предназначенный для представления табличных данных. Каждая строка в файле CSV соответствует строке в таблице. На одной линии поля разделяются символом запятой, каждое поле принадлежит одному столбцу таблицы. Разделителем может быть и другой символ, например, точка с запятой и т.д. Значения, содержащие зарезервированные символы (двойная кавычка, запятая, точка с запятой, новая строка) обрамляются двойными кавычками («»), иногда двойными кавычками обрамляют и текстовые значения. Т.к. разделители столбцов и строк могут быть разными, так же может быть разной кодировка текстового файла и обрамление двойными кавычками, то все это усложняет перенос данных из одних программ в другие, несмотря на всю простоту реализации поддержки CSV.

Таким образом, возникает ряд проблем при таком подходе:

1. Например, в CSV-файлах отсутствует значение для ID, которое во многих таблицах является первичным автоматически обновляемым ключом. В связи с этим могут возникнуть ошибки при импорте данных.



2. Необходимо привести данные к формату SQL Server (например, очередность полей в CSV-файле и в таблице SQL Server должна быть одинакова) и осуществить ручную обработку некорректных записей, что при размере таблиц в несколько сотен или тысяч записей является трудоемкой задачей.

3. Нужно устранить все возникающие исключения такие, как дублирование ключей, недопустимость NULL значений, невозможность приведения типов и т. д.

Лишь только после устранения всех исключений можно приступить к импорту файлов в MS SQL Server, что также является не простой задачей, поскольку CSV-файлы с разделителями-запятыми не поддерживаются в операциях массового импорта SQL Server. Однако в некоторых случаях CSV-файл можно использовать в качестве файла данных для массового импорта данных в SQL Server. При этом CSV-файл, который можно использовать в качестве файла данных для массового импорта, должен соответствовать следующим условиям:

- поля данных не должны содержать признак конца поля;
- или никакие, или все значения в полях данных должны заключаться в кавычки (").

После преобразования CSV-файла к виду, удовлетворяющего всем перечисленным ограничениям, можно переходить к непосредственному импорту данных, который можно реализовать следующими способами:

1. Можно использовать bsr-программы. Это программа командной строки (Bsr.exe), массово экспортирующая и импортирующая данные и создающая файлы форматирования. Файл форматирования служит для хранения сведений о форматировании для каждого поля в файле данных. Он также может содержать сведения о соответствующей таблице SQL Server. Файл форматирования может быть использован с целью предоставления всех сведений о форматировании, необходимых для массового экспорта данных из экземпляра SQL Server и массового импорта данных в него.

2. BULK INSERT инструкция. Инструкция Transact-SQL, импортирующая данные непосредственно из файла данных в таблицу базы данных или несекционированное представление.

3. Инструкция INSERT ... SELECT * FROM OPENROWSET(BULK...). Это инструкция Transact-SQL, использующая поставщик больших наборов строк OPENROWSET для массового импорта данных в таблицу SQL Server с помощью функции OPENROWSET(BULK...), применяющейся для выборки данных в предложение INSERT [2].

2. Импорт dbf-таблиц в базу данных MySQL с использованием базовых функции PHP.

Веб-приложение или сайт, разработанные на связке PHP+MySQL, имеют ряд значительных преимуществ. Для начала проведем сравнительный анализ достоинств и недостатков при использовании связки PHP+MySQL.

Практический характер применения PHP обусловлен следующими важными преимуществами:

1. Традиционностью. Язык PHP схож с языками C и Perl. Многие его конструкции заимствованы из этих языков. PHP сочетает в себе все достоинства C и Perl, и он специально нацелен на работу в Интернете. Все это снижает начальные усилия по его изучению.



2. Простотой. PHP — язык, который может быть встроен непосредственно в html-код веб-страниц, которые, в свою очередь, корректно обрабатываются PHP-интерпретатором. PHP можно использовать для:

- написания CGI-сценариев (CGI-скриптов), тем самым избавившись от множества неудобных операторов вывода текста.
- формирования HTML-документов, избавившись от множества вызовов внешних сценариев.

3. Эффективностью. «Движок» PHP по своей сути не является ни компилятором, ни интерпретатором. Он представляет собой транслирующий интерпретатор. Такое устройство «движка» PHP позволяет обрабатывать сценарии с достаточно высокой скоростью. Но чтобы ни делали разработчики PHP, откомпилированные исполняемые файлы работают значительно быстрее – в десятки и в сотни раз. Однако производительности PHP вполне достаточно для создания серьезных web-приложений.

4. Безопасностью. PHP предоставляет в распоряжение разработчиков и администраторов гибкие и эффективные средства безопасности, которые условно делятся на две категории: средства системного уровня и средства уровня приложения.

5. Гибкостью. PHP является встраиваемым (embedded) языком, поэтому он отличается высокой гибкостью по отношению к потребностям разработчика. В большинстве случаев PHP обычно рекомендуется использовать в сочетании с HTML, однако он хорошо интегрируется и в JavaScript, WML, XML и другие языки. Также не стоит проблемы с зависимостью от браузеров, поскольку перед отправкой клиенту сценарии PHP полностью компилируются на стороне сервера.

6. Бесплатным распространением. PHP распространяется бесплатно по стратегии OpenSource (с открытыми исходными кодами). Это является важным фактором в развитии проекта PHP.

MySQL – это одна из самых популярных и самых распространенных СУБД (система управления базами данных) в Интернете. Она не предназначена для работы с большими объемами информации, но ее применение идеально для интернет-сайтов, как небольших, так и достаточно крупных. MySQL отличаются хорошей скоростью работы, надежностью, гибкостью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку PHP. Немаловажным фактором является ее бесплатность. MySQL распространяется на условиях общей лицензии GNU (GPL, GNU PublicLicense).

Приложение на PHP, использующее для хранения информации базу данных (в частности MySQL) всегда работает быстрее приложения, построенного на файлах. Дело в том, что базы данных написаны на языке C++, и написать на PHP программу, которая работала бы с жёстким диском эффективнее базы данных - задача неразрешимая по определению, поскольку программы на PHP в принципе работают медленнее, чем программы на C++, так как PHP - интерпретатор, а C++ - компилятор [3,4].

Таким образом, основное достоинство базы данных заключается в том, что она берёт на себя всю работу с жёстким диском и делает это очень эффективно.



АПРОБАЦИЯ МЕТОДА ДОСТУПА К БД

Для реализации задачи исследования воспользуемся методом импорта dbf-таблиц в базу данных MySQL. Для этого будем использовать OpenServer в качестве локального сервера. OpenServer – это наилучший на сегодняшний день инструмент для веб-разработки под ОС Windows. Он представляет собой портативный локальный WAMP/WNMP сервер, имеющий многофункциональную управляющую программу и большой выбор подключаемых компонентов. Кроме того, OpenServer – это портативный сервер, не требующий установки, т.е. его можно запускать с любого съемного запоминающего устройства [5].

Процесс импорта dbf-таблиц в базу данных MySQL и на html-страницу происходит следующим образом:

1. Скрипт index.php создает html-страницу для загрузки файла dbf (рис. 3);

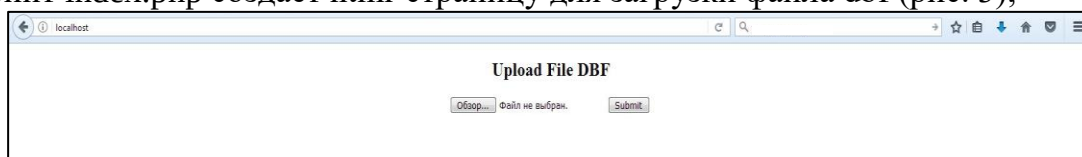


Рис. 3 – Страница загрузки dbf-файла

2. При выборе файла на этой странице вызывается скрипт upload.php или read_dbf.php (они могут работать и параллельно):

- upload.php предназначен для считывания dbf-файла, выбранного на странице загрузки, в базу данных MySQL;
- read_dbf считывает информацию из dbf файла в табличном виде на страницу html (рис. 4).

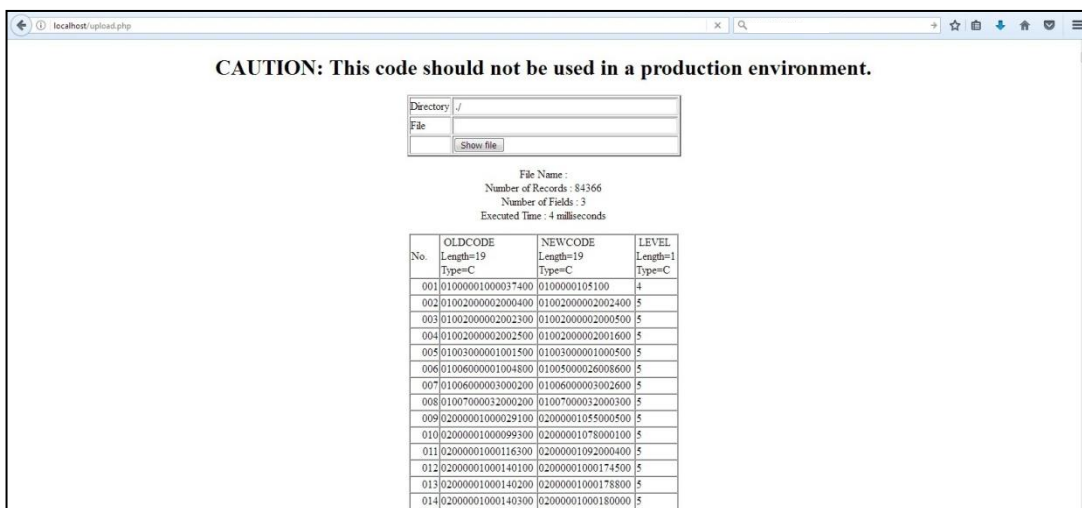


Рис. 4 – Html-страница с результатом загрузки dbf-файла

Оба скрипта upload.php и read_dbf.php для считывания информации из dbf-файла базы данных используют класс dbf_class, реализованный на основе базовых функций PHP. Он позволяет считывать данные (имена полей и соответствующие значения в строках) из файла базы данных и сохранять их в переменные массива



класса. Этот класс не требует никаких дополнительных расширений PHP для обработки dbf-файлов базы данных. Его можно использовать для перевода файла dbf в любой формат.

Список цитируемой литературы

1. Клиент-сервер – [Электронный ресурс]. – Режим доступа: <http://dic.academic.ru/dic.nsf/ruwiki/970166>(дата обращения: 22.05.2017г.)
2. Массовый импорт и экспорт данных (SQLServer)–[Электронный ресурс]. – Режим доступа: [https://technet.microsoft.com/ru-ru/library/ms175937\(v=sql.110\).aspx](https://technet.microsoft.com/ru-ru/library/ms175937(v=sql.110).aspx)(дата обращения: 25.05.2017г.)
3. Колисниченко, Д. Н. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д. Н. Колисниченко. – Санкт-Петербург, БХВ-Петербург, 2010. – 546 с.
4. Харрис Э. PHP/MySQL для начинающих / Энди Харрис. – Москва, КУДИЦ-Образ, 2005. – 384с.
5. OpenServer – [Электронный ресурс]. Режим доступа: - <http://ospanel.io/>(дата обращения: 29.05.2017г.)

© Л.И. Сасова, 2019