

**И.А.Смотров, Юсеф Момани, Р.В.Мальчева**  
Донецкий Национальный Технический Университет, г.Донецк  
кафедра компьютерной инженерии

## **РАЗРАБОТКА СТРУКТУРЫ БАЗЫ ДАННЫХ ДЛЯ ПОСТРОЕНИЯ МНОГОЯЗЫЧНОГО WEB-ПРИЛОЖЕНИЯ**

### *Аннотация*

*Смотров И.А., Момани Юсеф, Мальчева Р.В. Разработка структуры базы данных для построения конкретного web-приложения. Приведена классификация баз данных методом обзора существующих технологий. Определены предназначение и необходимый тип базы данных для реализации конкретного web-приложения. Определен подход формирования требований к базе данных для реализации конкретного приложения. Реализована модель базы данных.*

*Ключевые слова:* база данных, web-приложение, СУБД, сущность, домен, многоязычность, реляционная модель

**Постановка проблемы.** В последнее время web-приложения получили все большее распространение. Одним из важных преимуществ таких приложений является то, что пользователь не зависим от конкретной операционной системы. Для отображения web-приложения необходим только браузер. Именно поэтому они уверенно занимают свою нишу и играют важную роль во многих сферах жизни человека.

К современному web-приложению применяется все большее количество требований, в числе которых и многоязычного содержимого.

В web-приложениях могут быть динамические и статические страницы, информация о которых хранится в базе данных и в теле страницы соответственно. Создание многоязычного web-приложения с динамическим содержимым имеет свои особенности, заключающиеся, прежде всего, в создании оптимальной структуры базы данных, которая должна быть легко администрируема.

На данный момент есть несколько различных вариантов структур баз данных. Но каждый из них имеет свои недостатки. Так, например, создание таблицы с дублируемыми на разных языках полями, приводит к тому, что таблица имеет большой размер, содержимое не удобно редактировать и с увеличением количества языков увеличивается и количество полей в таблице.

Дублирование полей приводит к увеличению нагрузки на сервер и непременно сказывается на скорости загрузки страницы и выполнения запросов. Поэтому, необходимо разработать наиболее оптимальную структуру базы данных для многоязычного приложения.

**Анализ литературы.** Выполнен обзор основных типов баз данных, таких как иерархические, объектно-ориентированные, объектно-реляционные, реляционные, сетевые, функциональные и систем управления базами данных, таких как файл-серверные, клиент-серверные, встраиваемые системы.

На основе проделанного анализа выбрана СУБД, определены требования к структуре базы данных. Основными отличиями разрабатываемой базы данных будут легкость редактирования таблиц, полная связь переводов с соответствующей страницей.

**Цель статьи** – выбор типа доступа и разработка оптимальной структуры базы данных для реализации многоязычного web-приложения с динамическим содержимым.

**Классификация баз данных и систем управления.** База данных — представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов и т.д.), систематизированных таким образом, чтобы эти материалы могли быть найлены и обработаны при помощи компьютера [1].

Наибольшее распространение в процессе проектирования получили реляционные базы данных. Сама же реляционная модель основывается на математических принципах, вытекающих непосредственно из теории множеств и логики предикатов. Эти принципы впервые были применены в области моделирования данных в конце 60-х гг. доктором Е.Ф. Коддом, в то время работавшим в IBM, а впервые опубликованы в 1970 г [2]. Реляционная модель определяет способ представления данных (структуру данных), методы защиты данных (целостность данных), а также операции, выполняемые с данными (манипулирование данными). Реляционная модель – не единственный метод хранения и манипулирования данными. Существуют альтернативные варианты: иерархические, объектно-ориентированные, объектно-реляционные, реляционные, сетевые, функциональные. У каждой из них свои преимущества для решения задач определенного типа. Например, применение реляционной модели в обработке данных с иерархической организацией недостаточно хорошо изучено, для решения подобных задач используют специально созданную звездообразную модель данных. Тем не менее, гибкость и эффективность реляционной модели делают ее наиболее популярным инструментом для разработки баз данных [1].

Данные в базе хранятся в виде двумерных таблиц. Для добавления или редактирования содержимого в таблицах, а также их создания, используются СУБД - системы управления базами данных. СУБД проявляет себя как мощный инструмент тогда, когда информация, которую необходимо организовать и которой уже необходимо манипулировать, становится объемной и сложной, вследствие чего записи становятся непонятными и трудно обрабатываемыми вручную. Конечно, базы данных могут

использоваться большими корпорациями, обрабатывающими миллионы транзакций в день. Но СУБД может также потребоваться для маломасштабных операций, которые обрабатывает один человек в личных целях. Можно привести множество примеров ситуаций, когда СУБД будет использоваться даже в том случае, когда объемы информации не возросли до огромных размеров [3].

СУБД классифицируются по способу доступа к базе данных. Различают файл-серверные (Microsoft Access, Paradox, dBase, FoxPro, Visual FoxPro), клиент-серверные (Oracle, Firebird, Interbase, MS SQL Server, Sybase Adaptive Server Enterprise, PostgreSQL, MySQL), встраиваемые системы (SQLite, Microsoft SQL Server Compact).

В файл-серверных СУБД файлы данных располагаются на сервере. СУБД располагается на каждом клиентском компьютере. Доступ к данным осуществляется через локальную сеть. На данный момент файл-серверная технология считается устаревшей, а её использование в крупных информационных системах — недостатком.

Клиент-серверная СУБД располагается на сервере вместе с БД и осуществляет доступ к БД непосредственно, а запросы клиента к серверу выполняет централизованно. Основные преимущества таких систем: потенциально более низкая загрузка локальной сети; удобство централизованного управления; удобство обеспечения надёжности, высокой доступности и высокой безопасности.

Встраиваемая СУБД предназначена для локального хранения данных своего приложения и не рассчитана на коллективное использование в сети. Физически встраиваемая СУБД чаще всего реализована в виде подключаемой библиотеки. Доступ к данным со стороны приложения может происходить либо через SQL, либо через специальные программные интерфейсы.

**Решение задачи и результаты исследований.** Основным требованием к разрабатываемой базе данных web-сайта является отсутствие дублирования полей и полная связь страниц с переводами, что предопределяет использование иерархических структур на всех уровнях базы данных.

Для разработки использована клиент-серверная СУБД MySQL, ввиду своей доступности, надежности и функциональности.

База данных содержит таблицы, каждая из которых является сущностью. Сущность это нечто такое, о чем нужно хранить информацию в разрабатываемой системе. Каждая таблица в свою очередь имеет свои атрибуты, а те в свою очередь домены [2].

Так есть сущности «section», «section\_i18n», которые описывает раздел сайта. Различие таблиц в том, что первая хранит общую, неперебиваемую информацию о разделе – ссылку (link), картинку (img), уровень раздела (level), а вторая, с префиксом «\_i18n» - переводимые данные, такие как описание (description), название (title), язык перевода (lang). Структура данных сущностей

приведена на рис.1. Между таблицами установлена связь «one-to-many», т.е каждый раздел может иметь несколько переводов на другой язык. Таблицы связаны между собой уникальным ключом `section_id`, который соответствует номеру раздела в таблице «`section`».

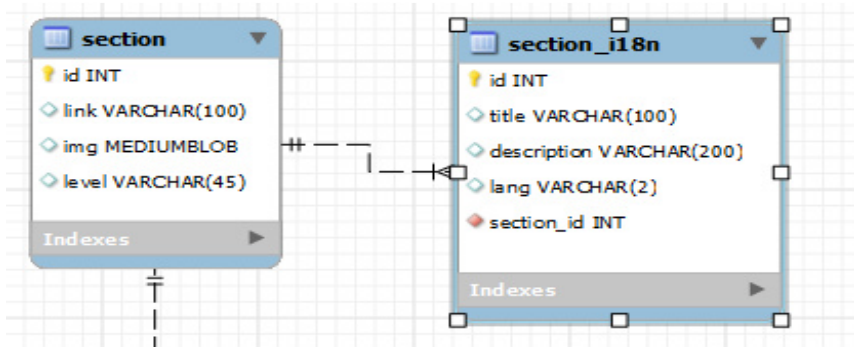


Рисунок 1 – Описание сущностей «`section`» и «`section_i18n`»

Создание таблиц происходит путем подобного SQL запроса в СУБД, который полностью описывает сущности и их атрибуты, а так же ключи и связи. Ниже приведен пример кода, создающего таблицу в БД:

```
CREATE TABLE IF NOT EXISTS `test`.`test_d` (
  `id` INT NOT NULL ,
  `item1` VARCHAR(100) NULL ,
  `item2` MEDIUMBLOB NULL ,
  .....
  .....
  .....
  `item n` VARCHAR(45) NULL ,
  PRIMARY KEY (`id`)
  FOREIGN KEY (`item n` )
  REFERENCES `test1`.`test_s` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION )
ENGINE = MyISAM
```

Аналогичным образом реализованы сущности «`page`» и «`page_i18n`», которые описывают страницу (рис.2). В таблице «`page`» есть ссылка, а в «`page_i18n`» атрибуты для описания переводимых параметров страницы, таких как название, содержимое, ключевые слова, метаданные, язык. Также как и в

случае с разделами, каждая страница может иметь несколько вариантов перевода, поэтому устанавливаем связь «one-to-many».

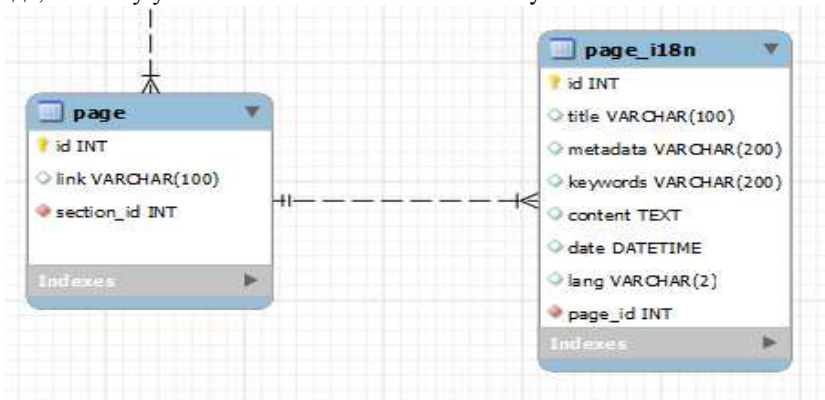


Рисунок 2 – Описание сущностей «page» и «page\_i18n»

Для реализации добавления и последующего вывода новостей были также реализованы дополнительные таблицы «news», «news\_i18n», также имеющие связь «one-to-many» (рис.3).

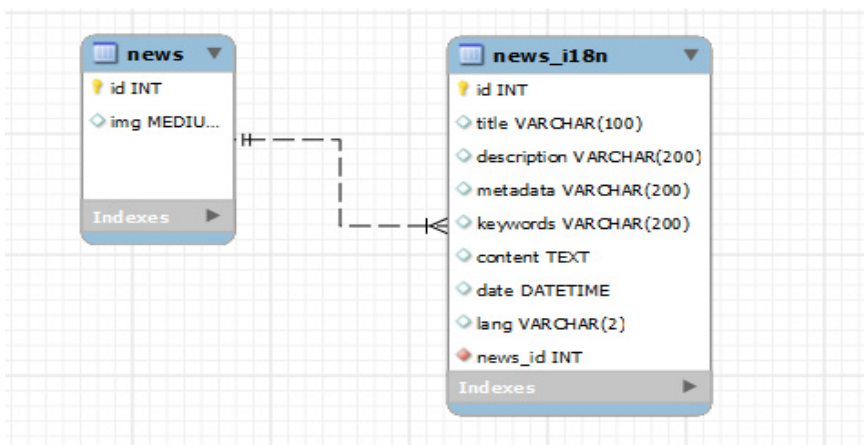


Рисунок 3 – Описание сущностей «news» и «news\_i18»

Каждый раздел может иметь несколько страниц, поэтому между «section» и «page» также предусмотрено отношение «one-to-many», связь по уникальному ключу page\_id. Это видно из общей физической структуры данных (рис.4).

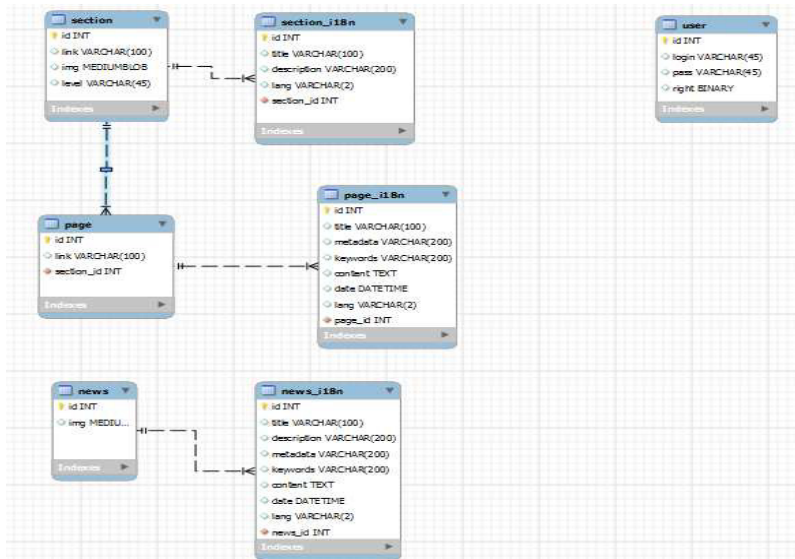


Рисунок 4 – Общая физическая структура данных

Таблица «user» относится к так называемой «системной» части приложения и содержит информацию о пользователях. В нашем случае это будут логин и пароль администратора.

**Выводы.** В результате работы были рассмотрены основные типы баз данных, системы управления, реализована структура базы данных для многоязычного web-приложения. База создана таким образом, чтобы можно было легко получать и редактировать динамическое содержимое web-страниц, что придаст заметную гибкость и удобство управления элементами данных.

Направлением дальнейших исследований является доработка и применение базы данных в реальном приложении, тестирование скорости запросов к базе данных, оптимизация таблиц средствами СУБД.

### Список литературы

1. Википедия [Electronic resource] / Интернет-ресурс. - Режим доступа : [www/ URL: http://ru.wikipedia.org](http://ru.wikipedia.org) - Загл. с экрана.
2. Райордан Р. Основы реляционных баз данных / Пер. с англ. М.: Издательско-торговый дом «Русская Редакция», 2001. — 384 с.: ил. ISBN 5-7502-0150-3
3. Дюбуа П. MySQL : Пер. с англ. : Уч. пос./ П. Дюбуа. — М.:Издательский дом "Вильямс", 2001. - 816 с. : ил.