

АҚПАРАТТЫҚ ЖӘНЕ КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР

УДК 378:004.652.4

Т. М. Маннапова¹, магистр технических наук, старший преподаватель

А.Ә. Құрманғалиева², магистрант

¹Западно-Казахстанского аграрно-технического университета им. Жангир хана, г.Уральск, РК

²Казахстанский университет инновационных и телекоммуникационных систем, г.Уральск, РК

ИСПОЛЬЗОВАНИЕ БАЗЫ ДАННЫХ MYSQL ДЛЯ СОЗДАНИЯ БАЗЫ ДАННЫХ РАСПИСАНИЯ В ВУЗЕ

Аннотация

В статье рассматриваются вопросы добавления данных в базу данных и их отображение в браузере с помощью языка PHP, освоение теоретических основ построения баз данных, перспективных моделей данных, различных средств проектирования баз данных, особенностей их физической организации, средств защиты баз данных и обеспечения целостности и сохранности данных. Были рассмотрены SQL-запросы к базе данных.

Ключевые слова: база данных, реляционная база данных, сервер MySQL, PHP, HTML, SQL, PhpMyAdmin, table.

Современный мир информационных технологий трудно представить себе без использования баз данных. Практически все системы в той или иной степени связаны с функциями долговременного хранения и обработки информации. Фактически использование информации становится фактором, определяющим эффективность любой сферы деятельности. Увеличились информационные потоки и повысились требования к скорости обработки данных, обработка информации требует применения наиболее перспективных компьютерных технологий [1].

Современные базы данных являются основой многочисленных информационных систем и информация, накопленная в них, является чрезвычайно ценным материалом. Интенсивно развиваются методы обработки баз данных, позволяющие извлечь из них дополнительные знания, обобщить имеющуюся информацию и обработать ее дополнительными способами. Базы данных в данной концепции выступают как хранилища информации.

Средства эффективного хранения и выборки больших объемов информации внесли огромный вклад в успешное развитие Интернета. Обычно для хранения информации используются базы данных. Работа таких известных сайтов, как Yahoo, Amazon и Ebay, в значительной степени зависит от надежности баз данных, хранящих громадные объемы информации. Поддержка баз данных ориентирована не только на интересы гигантских корпораций – в распоряжении web-программистов имеется несколько мощных реализаций баз данных, распространяемых по низкой цене.

В жизни мы часто сталкиваемся с необходимостью хранить какую-либо информацию, а потому часто имеем дело и с базами данных. Например, разработка технологии создания расписания учебного процесса для быстрого составления расписания в электронном виде и проверки ее эффективности в реальном творческом проекте. Все это своего рода базы данных. Ну а раз это базы данных, то посмотрим, как в них хранятся данные. Например, расписания занятий на факультетах университета представляет собой таблицу (Таблица 1).

В этой таблице данные – это собственно ФИО преподавателя, номера кабинета, день недели, дисциплина, группа.

Таблица 1 – Пример базы данных: Расписания факультетов университета

ФИО преподавателя	Номер кабинета	День недели	Дисциплина	Группа
Ильин Федор Иванович	202	понедельник	Математика	ИС-12
Петров Александр Иванович	405	вторник	Физика	НГД-12

В данном примере только две записи, а если в таблице будет несколько тысяч записей, то возможно где-то произойдет ошибка (например, опечатка в дисциплине). Тяжело будет найти и исправить эту ошибку вручную. Для исправления нужно воспользоваться какими-то средствами автоматизации. Для управления большим количеством данных программисты (не без помощи математиков) придумали системы управления базами данных (СУБД). По сравнению с текстовыми базами данных электронные СУБД имеют огромное число преимуществ: от возможности быстрого поиска информации, взаимосвязи данных между собой до использования этих данных в различных прикладных программах и одновременного доступа к данным нескольких пользователей.

База данных – это совокупность связанных данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования, независимая от прикладных программ. База данных является информационной моделью предметной области. Обращение к базам данных осуществляется с помощью системы управления базами данных [3].

Способов хранения данных существует множество. Наиболее популярные – объектная и реляционная модели данных.

Реляционная база данных представляет собой набор таблиц (таких таблиц как приведено выше), связанных между собой. Строка в таблице соответствует сущности реального мира (в приведенном выше примере это готовое расписание).

Автором реляционной модели считается Э. Кодд, который первым предложил использовать для обработки данных аппарат теории множеств (объединение, пересечение, разность, декартово произведение) и показал, что любое представление данных сводится к совокупности двумерных таблиц особого вида, известного в математике как отношение.

При работе с MySQL важно учесть аспекты работы с реляционными базами данных, которые имеют два важных понятия: ключи и индексирование.

Первичный ключ (primary key, PK) – минимальный набор полей, уникально идентифицирующий запись в таблице. Значит, первичный ключ – это в первую очередь набор полей таблицы, во-вторых, каждый набор значений этих полей должен определять единственную запись (строку) в таблице и, в-третьих, этот набор полей должен быть минимальным из всех обладающих таким же свойством. Поскольку первичный ключ определяет только одну уникальную запись, то никакие две записи таблицы не могут иметь одинаковых значений первичного ключа [4].

В любом случае проблема создания первичного ключа ложится на плечи того, кто проектирует базу данных. Решением этой проблемы может стать либо выделение характеристик, которые естественным образом определяют запись в таблице, либо создание дополнительного поля, предназначенного именно для однозначной идентификации записей в таблице. Кроме однозначной идентификации записи, первичные ключи используются для организации связей с другими таблицами.

Одна из основных задач, возникающих при работе с базами данных, – это задача поиска. При этом, поскольку информации в базе данных, как правило, содержится много, перед программистами встает задача не просто поиска, а эффективного поиска, т.е. поиска за сравнительно небольшое время и с достаточной точностью. Для этого производят индексирование некоторых полей таблицы [4].

Приведем пример создания базы данных для составления расписания в ВУЗе. Для этого сначала разрабатывается инфологическая модель, которая является ядром системы проектирования. Она содержит всю необходимую информацию для проектирования приложения.

Для описания инфологической модели данных можно выделить шесть объектов, которые участвуют в обработке информации: «Кабинеты», «День недели и время», «Преподаватели», «Предметы», «Группы», «Расписание». Каждый объект обладает своими свойствами.

Объект «Расписание» является связывающим звеном между всеми выше перечисленными объектами. Он обладает свойствами: «Идентификатор расписания», «Идентификатор кабинета», «Идентификатор дня», «Идентификатор преподавателя», «Идентификатор предмета», «Идентификатор группы».

В каждой таблице присутствует уникальное идентификационное поле, которое позволяет однозначно идентифицировать запись таблицы. Такая структура исключает избыточность данных, позволяет ускорить отбор записей по условию, исключает вероятность конфликтов при совместной работе нескольких пользователей.

Одной из основных таблиц является таблица "cabinets". В ней хранится вся информация о кабинетах учебного заведения. Таблица содержит поля: cab_id – идентификатор кабинетов (ключевое поле. По полю cab_id данная таблица связана с таблицей «ready_schedule».

По такому же принципу создаем таблицы: «days» с ключевым полем day_id, «groups» с ключевым полем group_id, «names» с ключевым полем name_id, «subjects» с ключевым полем subject_id.

В таблице «ready_schedule» формируется расписание, в котором данные берутся из вышесказанных таблиц. Поле id_ready_schedule является ключевым, cab_id – идентификатор кабинета, day_id – идентификатор дня, name_id – идентификатор преподавателя, subject_id – идентификатор предмета, group_id – идентификатор группы.

NOT NULL означает, что все строки таблицы должны иметь значение в этом атрибуте. Если NOT NULL не указано, поле может быть пустым (NULL).

AUTO_INCREMENT – специальная возможность MySQL, которую можно задействовать в числовых столбцах. Если при вставке строк в таблицу оставлять такое поле пустым, MySQL автоматически генерирует уникальное значение идентификатора. Это значение будет на единицу больше максимального значения, уже существующего в столбце. В каждой таблице может быть не больше одного такого поля. Столбцы с AUTO_INCREMENT должны быть проиндексированными.

Структура таблицы с описанием типов полей приведена в таблице 2.

Таблица 2 – Структура таблицы «ready_schedule»

№	Описание	Поле	Тип данных	Дополнительные параметры
1	Идентификатор расписания	id_ready_schedule	int(11)	AUTO_INCREMENT
2	Идентификатор кабинета	cab_id	int(11)	NOT NULL
3	Идентификатор дня	day_id	int(11)	NOT NULL
4	Идентификатор преподавателя	name_id	int(11)	NOT NULL
5	Идентификатор дисциплины	subject_id	int(11)	NOT NULL
6	Идентификатор группы	group_id	int(11)	NOT NULL

В реляционных СУБД данные организуются в виде набора взаимосвязанных таблиц. Связи между таблицами реализуются в виде ссылок на данные других таблиц. Таблицу можно представить себе как двухмерный массив, в котором расположение каждого элемента характеризуется определенными значениями строки и столбца.

Использовать индексы полезно для быстрого поиска строк с указанным значением одного столбца. Без индекса чтение таблицы осуществляется по всей таблице, начиная с первой записи, пока не будут найдены соответствующие строки. Чем больше таблица, тем больше накладные расходы. Если же таблица содержит индекс по рассматриваемым столбцам, то база

данных может быстро определить позицию для поиска в середине файла данных без просмотра всех данных. Это происходит потому, что база данных помещает проиндексированные поля поближе в памяти, так, чтобы можно было побыстрее найти их значения. Для таблицы, содержащей 1000 строк, это будет как минимум в 100 раз быстрее по сравнению с последовательным перебором всех записей. Однако в случае, когда необходим доступ почти ко всем 1000 строкам, быстрее будет последовательное чтение, так как при этом не требуется операций поиска по диску. Так что иногда индексы бывают только помехой. Например, если копируется большой объем данных в таблицу, то лучше не иметь никаких индексов. Однако в некоторых случаях требуется задействовать сразу несколько индексов.

В MySQL, то там существует три вида индексов: PRIMARY, UNIQUE, и INDEX, а слово ключ (KEY) используется как синоним слова индекс (INDEX). Все индексы хранятся в памяти в виде B-деревьев. Если вы собираетесь пользоваться базой данных на чужом сервере, то его администратор проделает все описанные выше действия за вас, т.е. все настроит и создаст пользователя и базу данных.

Примеры реляционных СУБД: Mysql, PostgreSQL. В основу объектной модели положена концепция объектно-ориентированного программирования, в которой данные представляются в виде набора объектов и классов, связанных между собой родственными отношениями, а работа с объектами осуществляется с помощью скрытых (инкапсулированных) в них методов. Подробнее рассмотрим СУБД Mysql.

Первоначально сервер MySQL разрабатывался для управления большими базами данных с целью обеспечить более высокую скорость работы по сравнению с существующими на тот момент аналогами. И вот уже в течение нескольких лет данный сервер успешно используется в условиях промышленной эксплуатации с высокими требованиями. Несмотря на то, что MySQL постоянно совершенствуется, он уже сегодня обеспечивает широкий спектр полезных функций. Благодаря своей доступности, скорости и безопасности MySQL очень хорошо подходит для доступа к базам данных по Интернету[2].

Работать с Mysql можно не только в текстовом режиме, но и в графическом. Существует очень популярный визуальный интерфейс (написанный на PHP) для работы с этой СУБД. Называется он PhpMyAdmin. Этот интерфейс позволяет значительно упростить работу с базами данных в Mysql. PhpMyAdmin позволяет пользоваться всеми достоинствами браузера, включая прокрутку изображения, если оно не умещается на экран. Многие из базовых SQL-функций работы с данными в PhpMyAdmin сведены к интуитивно понятным интерфейсам и действиям, напоминающим переход по ссылкам в Internet.

В вышеприведенном примере создания базы данных для составления расписания в ВУЗе на основании анализа предметной области и построенной инфологической модели была разработана реляционная база данных "dis_rabota" (рисунок 1). Для реляционной базы данных проектирование физической структуры заключается в том, чтобы разбить всю информацию по таблицам, а также определить состав полей для каждой из этих таблиц и установить связи между таблицами.

Если База данных (СУБД) проектируется в MySQL, можно программировать на языке PHP, структуру сайта разработать на HTML. Разработка web-страниц на чистом HTML является web-программированием, ведь при просмотре страницы браузер фактически исполняет код HTML, форматируя текст согласно инструкциям этого языка. В настоящее время под web-программированием понимают создание CGI-приложений и использование на web-странице технологий JavaScript и VBScript для достижения сложных эффектов.

Разработчикам Web-приложений нет необходимости говорить, что web-страницы – это не только текст и картинки. Достойный внимания сайт должен поддерживать некоторый уровень интерактивности с пользователем: поиск информации, продажа продуктов, конференции и т.п.

Синтаксис PHP очень похож на синтаксис C или Perl. Люди, знакомые с программированием, очень быстро смогут начать писать программы на PHP. В этом языке нет строгой типизации данных и нет необходимости в действиях по выделению/освобождению памяти. Программы, написанные на PHP, достаточно легко читаемы. Написанный PHP – код легко зрительно прочитать и понять.

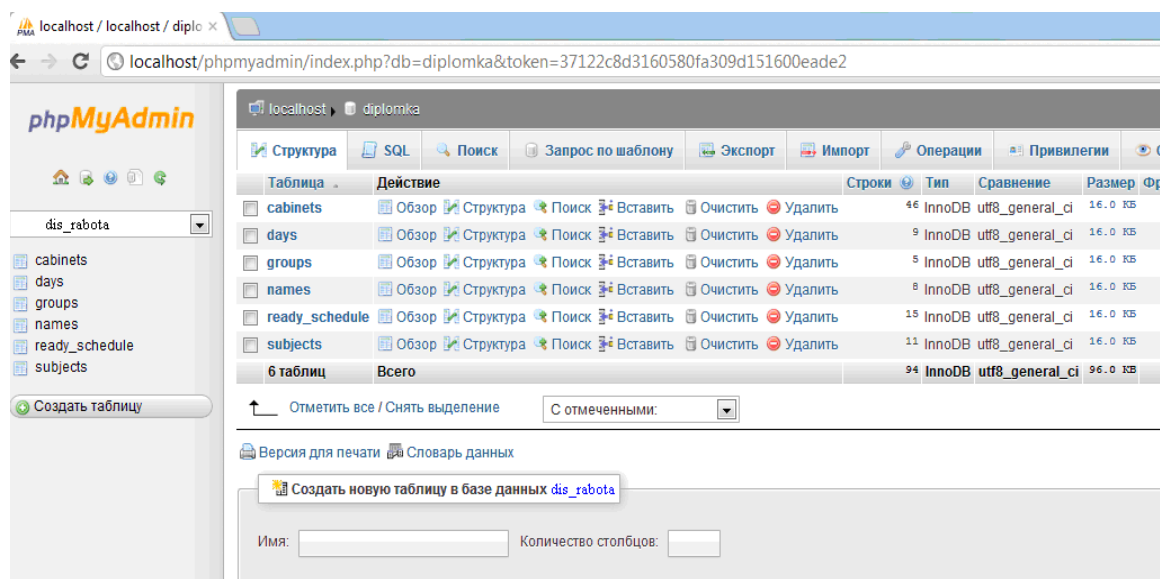


Рисунок 1 – Просмотр таблиц базы данных dis_rabota с использованием программы phpMyAdmin

Благодаря хорошим характеристикам и обширному набору стандартных интерфейсных функций, очень простых в использовании, MySQL стала самым популярным средством для работы с базами данных в РНР. Для использования функций работы с базой данных MySQL следует скомпилировать РНР с поддержкой этого расширения. В последнее время наиболее распространенные расширения и соответственно их функции изначально включают в состав РНР так, чтобы с ними можно работать без каких бы то ни было дополнительных настроек интерпретатора.

MySQL – компактный многопоточный сервер баз данных. MySQL характеризуется большой скоростью, устойчивостью и легкостью в использовании. MySQL был разработан компанией ТсХ для внутренних нужд, которые заключались в быстрой обработке очень больших баз данных. MySQL использует с 1996 года на сервере с более чем 40 БД, которые содержат 10,000 таблиц, из которых более чем 500 имеют более 7 миллионов строк. MySQL является идеальным решением для малых и средних приложений. Исходники сервера компилируются на множестве платформ. Наиболее полно возможности сервера проявляются на Unix-серверах, где есть поддержка многопоточности, что дает значительный прирост производительности.

MySQL – это реляционная система управления базами данных. То есть данные в ее базах хранятся в виде логически связанных между собой таблиц, доступ к которым осуществляется с помощью языка запросов SQL. SQL обычно описывается как стандартный язык, используемый для взаимодействия с реляционными базами данных. Однако SQL не является языком программирования, как C, C++ или РНР. Скорее, это интерфейсное средство для выполнения различных операций с базами данных, предоставляющее в распоряжение пользователя стандартный набор команд. Возможности SQL не ограничиваются выборкой данных из базы.

SQL – язык структурированных запросов, основной задачей которого является предоставление простого способа считывания и записи информации в базу данных. Итак, простейшая схема работы с базой данных выглядит примерно так (рисунок 2):

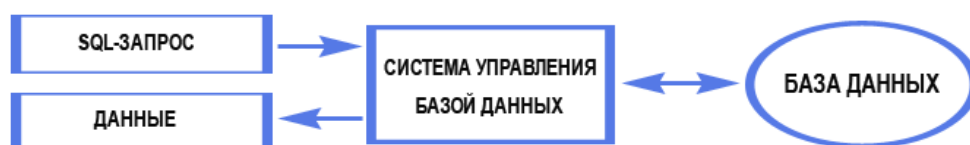


Рисунок 2 – Схема работы с базой данных

По характеру использования СУБД делят на однопользовательские (предназначенные для создания и использования БД на персональном компьютере) и многопользовательские (предназначенные для работы с единой БД нескольких компьютеров, объединенных в локальные сети). Вообще деление по характеру использования можно представить следующей схемой (рисунок 3):

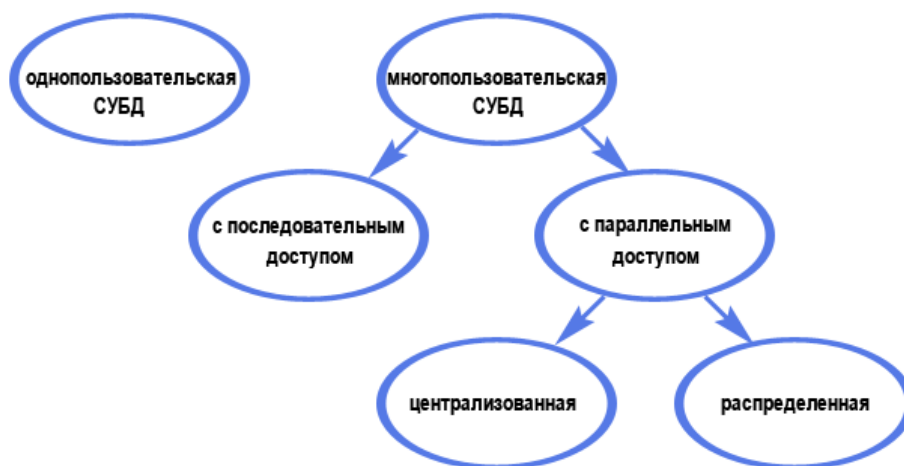


Рисунок 3 – СУБД по характеру использования

На сегодняшний день число используемых СУБД исчисляется десятками. Наиболее известные однопользовательские СУБД – Microsoft Visual FoxPro и Access, многопользовательские – MS SQL Server, Oracle и MySQL.

В текстовом режиме работа с базой данных выглядит просто как ввод команд в командную строку, а результаты выборки возвращаются в виде своеобразных таблиц, поля в которых налегают друг на друга, если данные не помещаются на экран.

Прежде чем что-либо делать с данными, нужно создать таблицы, в которых эти данные будут храниться, научиться изменять структуру этих таблиц и удалять их, если потребуется. Для этого в языке SQL существуют операторы CREATE TABLE, ALTER TABLE и DROP TABLE.

В дистрибутив PHP входит расширение, содержащее встроенные функции для работы с базой данных MySQL. Для решения задач построения web-интерфейсов с целью отображения и наполнения базы данных. Возникает вопрос, зачем строить такие интерфейсы? Для того чтобы вносить информацию в базу данных и просматривать ее содержимое могли люди, не знакомые с языком запросов SQL. При работе с web-интерфейсом для добавления информации в базу данных человеку нужно просто ввести эти данные в html-форму и отправить их на сервер, а наш скрипт сделает все остальное. А для просмотра содержимого таблиц достаточно просто щелкнуть по ссылке и зайти на нужную страницу.

В базе данных, которые мы использовали для составления расписания в ВУЗе имеется таблица. Чтобы построить интерфейс для добавления информации в эту таблицу, нужно ее структуру (т.е. набор ее полей) отобразить в html-форму. Для отображения таблицы в html-форму нужно решить следующие подзадачи:

- установка соединения с БД;
- выбор рабочей БД;
- получение списка полей таблицы;
- отображение полей в html-форму.

После этого данные, введенные в форму, нужно записать в базу данных.

После установки соединения нужно выбрать базу данных, с которой будем работать. Наши данные хранятся в базе данных dis_rabota. В MySQL выбор базы данных осуществляется с помощью команды use: Mysql> dis_rabota.

Можно сделать вывод, что для большинства Интернет-проектов возможностей СУБД MySQL вполне достаточно. Благодаря следующим преимуществам MySQL является самой подходящей СУБД в сфере Интернет:

Быстродействие. Благодаря внутреннему механизму многопоточности быстродействие MySQL весьма высоко.

Безопасность. Довольно высокий уровень безопасности обеспечивается благодаря базе данных mysql, создающейся при установке пакета и содержащей пять таблиц. При помощи этих таблиц можно описать, какой пользователь из какого домена с какой таблицей может работать и какие команды он может применять. Пароли, хранящиеся в базе данных, можно зашифровать при помощи встроенной в MySQL функции password().

Лицензия. Программа для некоммерческих целей распространяется бесплатно.

Открытость кода. Благодаря этому вы сможете сами добавлять в пакет нужные функции, расширяя его функциональность так, как вам требуется.

Надежность. MySQL довольно стабилен и его трудно вывести из строя.

СПИСОК ЛИТЕРАТУРЫ

- 1 Пуха Ю. Объектные технологии построения распределенных информационных систем / Ю. Пуха. – 1997. – 156 с.
- 2 Русаков М. НР и MySQL с Нуля до Гуру / М. Русаков. – 2012. – 205 с.
- 3 К. Дж. Дейт Введение в системы баз данных (седьмое издание). – М.- СПб - Киев. – 2001. – 132 с.
- 4 Кириллов В. В. Введение в реляционные базы данных / В. В. Кириллов, Г.Ю.Громов. – СПб. – 2009. – 26 с.

ТҮЙІН

Бұл мақалада деректер базасына деректерді қосу және PHP тілі арқылы шолғышта олардың көрсетілуі, деректер базасының теориялық негіздерін дамыту, кеңейтілген деректер үлгілерін тандау, деректер базасын жобалаудың әр түрлі құралдары, деректер базасының физикалық ұйымдастыру ерекшеліктері, деректер базасын қорғау және деректер тұтастығын мен қауіпсіздігін қамтамасыз ету қарастырылуда. Деректер базасына SQL-сұранымдар қарастырылды.

RESUME

This article explains how to add data to the database and display them in the browser using the PHP language, the development of the theoretical foundations of databases, selection of prospective models of data, various means of database design, the characteristics of the physical organization of databases, protection of databases and integrity and data integrity. SQL-query the database have been considered.