

ному" и "один ко многим". Например, один клиент может сделать несколько заказов, однако в то же время один заказ приходит от одного клиента. Так, в заказе создается ссылка на клиента.

- Связывающие таблицы, которые описывают отношения типа "многие ко многим" между двумя реальными объектами, например, между Orders и Books. Такие таблицы зачастую ассоциируются с некоторой реальной транзакцией.

Архитектура Web-баз данных

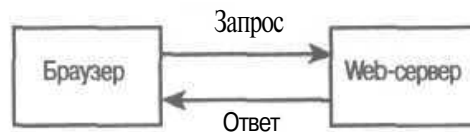
Теперь, когда рассмотрена внутренняя архитектура базы данных, пришло время взглянуть на внешнее построение системы Web-баз данных и обсудить методологию ее разработки.

Архитектура

Основная операция Web-сервера проиллюстрирована на рис. 7.8. Эта система состоит из двух объектов: Web-браузера и Web-сервера. Между ними должен существовать канал связи. Web-браузер посылает запрос на сервер, сервер отправляет обратно ответ. Для сервера, отправляющего обычные статические страницы, такая архитектура подходит. Архитектура же сайта, который включает в себя базу данных, несколько сложнее.

РИСУНОК 7.8

Отношение типа клиент/сервер между Web-браузером и Web-сервером требует наличия связи.



Приложения Web-баз данных, которые будут разрабатываться на страницах этой книги, наследуют глобальную структуру Web-баз данных, показанную на рис. 7.9. Большая часть этой структуры должна выглядеть знакомо.

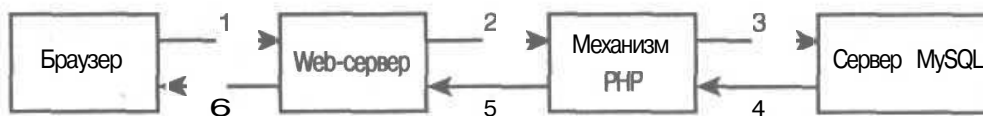


РИСУНОК 7.9 Базовая архитектура Web-баз данных включает в себя Web-браузер, Web-сервер, сценарный механизм и сервер баз данных.

Типичная транзакция Web-базы данных состоит из этапов, обозначенных цифрами на рис. 7.9. Мы рассмотрим их на примере магазина "Book-O-Rama".

1. Web-браузер пользователя отправляет **HTTP-запрос** определенной Web-страницы. Например, поиск в магазине "Book-O-Rama" всех книг, написанных Лорой Томсон (Laura Thomson), используя HTML-форму. Страница с результатами поиска называется **results.php**.
2. Web-сервер принимает запрос на results.php, получает файл и передает его механизму PHP на обработку.

3. Механизм PHP начинает синтаксический анализ сценария. В сценарии присутствует команда подключения к базе данных и выполнения запроса в ней (поиск книг). PHP открывает соединение с сервером MySQL и отправляет необходимый запрос.
4. Сервер MySQL принимает запрос в базу данных, обрабатывает его, а затем отправляет результаты — в данном случае, список книг — обратно в механизм PHP.
5. Механизм PHP завершает выполнение сценария, форматируя результаты запроса в виде HTML, после чего отправляет результаты в HTML-формате Web-серверу.
6. Web-сервер пересылает HTML в браузер, с помощью которого пользователь просматривает список необходимых книг.

Процесс этот, как правило, протекает вне зависимости от того, какой сценарный механизм и какой сервер баз данных используется. Зачастую программное обеспечение Web-сервера, механизм PHP и сервер баз данных находятся на одной машине. Правда, не менее часто сервер базы данных работает на другой машине. Это делается из соображений безопасности, увеличение объема или разделения потока. С точки зрения перспектив развития, в работе оба варианта одинаковы, однако в плане производительности второй вариант может оказаться более предпочтительным.

Дополнительная информация

В данной главе были раскрыты некоторые принципы проектирования реляционных баз данных. Если вы желаете глубже изучить теорию реляционных баз данных, можете прочесть книги реляционных гуру (например, Си Джей Дейта (C. J. Date)). Однако имейте в виду, что материал, который вы почерпнете, может оказаться по большей части теоретическим и не сразу найдет себе применение в коммерческих Web-разработках. Среднестатистическая база данных не должна быть особо сложной.

Что дальше

В следующей главе мы начнем настраивать базу данных MySQL. Сначала вы научитесь настраивать базу данных MySQL под Web и отправлять ей запросы, а после — отправлять запросы из PHP-кода.