

Э. В. ФУФАЕВ, Д. Э. ФУФАЕВ

РАЗРАБОТКА И ЭКСПЛУАТАЦИЯ УДАЛЕННЫХ БАЗ ДАННЫХ

УЧЕБНИК

Допущено

*Экспертным советом по профессиональному образованию
в качестве учебника для использования в учебном процессе
образовательных учреждений, реализующих программы
среднего профессионального образования по специальности
«Программное обеспечение вычислительной техники
и автоматизированных систем»*



Москва
Издательский центр «Академия»
2008

УДК 621.391(075.32)
ББК 32.81я723
Ф964

Рецензенты:

преподаватель Московского государственного колледжа информационных технологий *И. А. Кумскова*;
зам. декана факультета «Информатика и телекоммуникации» Московского государственного института электроники и математики, доц.,
канд. техн. наук *Д. П. Николаев*

Фуфаев Э. В.

Ф964 Разработка и эксплуатация удаленных баз данных : учебник для студ. сред. проф. образования / Э. В. Фуфаев, Д. Э. Фуфаев. — М. : Издательский центр «Академия», 2008. — 256 с.

ISBN 978-5-7695-3772-1

Даны теоретические основы и практические рекомендации по разработке и эксплуатации удаленных баз данных при создании информационных систем для различных задач управления. Рассмотрены современные технологии доступа к удаленным базам данных.

Для студентов средних профессиональных учебных заведений. Может быть полезен для специалистов, работающих в области управления производством и бизнесом.

УДК 621.391(075.32)
ББК 32.81я723

Оригинал-макет данного издания является собственностью Издательского центра «Академия», и его воспроизведение любым способом без согласия правообладателя запрещается

© Фуфаев Э. В., Фуфаев Д. Э., 2008

© Образовательно-издательский центр «Академия», 2008

ISBN 978-5-7695-3772-1

© Оформление. Издательский центр «Академия», 2008

ОГЛАВЛЕНИЕ

Предисловие	3
Введение	5

ЧАСТЬ I

Теоретические основы проектирования удаленных баз данных

Глава 1. Архитектуры удаленных баз данных	12
1.1. Термины и определения	12
1.2. Архитектуры клиент— сервер в технологии управления удаленными базами данных	13
1.3. Двухуровневые модели	16
1.4. Основные свойства распределенных баз данных	27
Глава 2. Принципы разработки и эксплуатации систем управления удаленными базами данных	31
2.1. CALS-технологии — основная концепция разработки удаленных баз данных	31
2.2. Принципы разработки многопользовательских информационных систем	33
2.3. Организация многопользовательских систем управления базами данных в локальных вычислительных сетях	35
2.4. Этапы проектирования многопользовательских баз данных	36
2.5. Администрирование баз данных	44

ЧАСТЬ II

Системы разработки и управления удаленными базами данных

Глава 3. Технологии разработки и управления базами данных средствами языка SQL	47
3.1. Назначение языка SQL	47
3.2. Основные правила записи операторов	48
3.3. Операторы манипулирования данными	48
Глава 4. Управление удаленными базами данных в системе SQL Server2000	53
4.1. Службы управления базами данных SQL Server2000	53
4.2. Системные базы данных SQL Server2000	57

Глава 5. Управление удаленными базами данных в системе Oracle	65
5.1. Основные понятия и термины	65
5.2. Типы пользователей	68
5.3. Физическая архитектура хранения данных	70
5.4. Транзакции	75
5.5. Обеспечение целостности данных	76
5.6. Создание триггеров и хранимых процедур	77
Глава 6. Технологии доступа к удаленным базам данных	80
6.1. Структура организации доступа к данным в трехуровневой архитектуре	80
6.2. Объектные модели доступа к удаленным базам данных	81
6.3. Монитор обработки транзакций	83
6.4. Универсальная стратегия доступа к данным ODBC	84
6.5. Технологии COM	87
6.6. Технологии ADO .NET	88
6.7. Технологии .NET FrameWork	90
6.8. Технологии CORBA	94
6.9. Технологии MIDAS	98

ЧАСТЬ III

Проектирование серверной части приложения баз данных

Глава 7. Методические основы проектирования серверной части приложения	102
Глава 8. Технологии проектирования серверной части приложения	109
8.1. Применение СУБД Access для разработки проекта удаленных баз данных	109
8.2. Создание серверного приложения преобразованием проекта базы данных формата Microsoft Access в формат SQL Server	111
8.3. Проектирование и модификация таблиц командами SQL	114
8.4. Создание пользовательских представлений	118
8.5. Разработка хранимых процедур	119
8.6. Разработка триггеров	129

ЧАСТЬ IV

Проектирование клиентской части приложения баз данных

Глава 9. Общие принципы проектирования клиентской части баз данных	132
9.1. Основные требования к разработке пользовательского интерфейса	132
9.2. Разработка пользовательского интерфейса средствами визуального проектирования MS Access	133

Глава 10. Разработка программ управления удаленными базами данных с применением операторов SQL	150
10.1. Внедрение операторов SQL в прикладные программы	150
10.2. Выполнение однострочных и многострочных запросов с помощью внедренных операторов SQL и курсоров	151
10.3. Модификация таблиц баз данных с помощью курсоров	153
Глава 11. Web-технологии в разработке удаленных баз данных	157
11.1. Введение в Интернет и среду WWW	157
11.2. Статические и динамические Web-страницы	161
11.3. Требования к интеграции удаленных баз данных со средой Web	162
11.4. Методы интеграции удаленных баз данных в среду Web	163
11.5. Генерация Web-страниц визуальными средствами Microsoft Access	165

ЧАСТЬ V

Администрирование и эксплуатация удаленных баз данных

Глава 12. Защита информации и управление доступом к данным	168
12.1. Основные проблемы и способы защиты баз данных	168
12.2. Технологические методы защиты информации	169
12.3. Дисковое хранилище с системой уничтожения данных	182
12.4. Программа для создания зашифрованной области на жестком диске DriveCrypt Plus Pack 3	186
12.5. Организационные рекомендации по обеспечению безопасности эксплуатации удаленных баз данных	195
Глава 13. Восстановление данных в критических ситуациях	199
13.1. Восстановление базы данных	199
13.2. Транзакции и восстановление	200
13.3. Управление буферами базы данных	202
13.4. Механизм резервного копирования	203

ЧАСТЬ VI

Постреляционные системы управления удаленными базами данных

Глава 14. Ориентация развития СУБД на расширенную реляционную модель	206
14.1. Основные направления совершенствования реляционных баз данных	206
14.2. Генерация систем баз данных, ориентированных на приложения	208
14.3. Оптимизация запросов, управляемых правилами	209

14.4. Поддержка динамической информации и темпоральных запросов	209
Глава 15. Объектно-ориентированные СУБД	211
15.1. Общие понятия объектно-ориентированного подхода к разработке СУБД	211
15.2. Объектно-ориентированные модели данных	214
15.3. Языки программирования объектно-ориентированных баз данных	215
Глава 16. Объектно-ориентированная СУБД Cache	219
16.1. Структура СУБД Cache	219
16.2. СУБД Cache и Web-технологии	222
16.3. Среда разработки приложений Visual Basic.NET	224
16.4. Многоплатформенный протокол передачи данных SOAP	225
Глава 17. Системы баз данных, основанные на правилах	229
17.1. Структура базы данных	229
17.2. Активные базы данных	230
17.3. Дедуктивные базы данных	230
Глава 18. Многопользовательские системы управления жизненным циклом продукции	233
18.1. Интегрированная информационная среда предприятия	233
18.2. Структура и состав интегрированной информационной среды предприятия	236
18.3. Управление интегрированной информационной средой предприятия	240
18.4. Управление качеством	241
18.5. Управление потоками работ	243
Список литературы	246

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРОЕКТИРОВАНИЯ УДАЛЕННЫХ БАЗ ДАННЫХ

ГЛАВА 1

АРХИТЕКТУРЫ УДАЛЕННЫХ БАЗ ДАННЫХ

1.1. Термины и определения

Системы управления удаленными (распределенными) базами данных — это СУБД (СУРБД), обеспечивающие возможность одновременного доступа к информации различным пользователям.

Рассмотрим термины, применяемые в системах управления распределенными базами данных.

Архитектура БД — организация взаимодействия аппаратных средств.

Виды архитектуры БД: клиент — сервер, двухуровневая и трехуровневая клиент-сервер, файл — сервер.

Архитектура ODBC (Open DataBase Connectivity) — открытый интерфейс доступа к базам данных, т. е. взаимодействие процессора (ядра) базы данных Jet с внешними источниками данных.

Модели данных — схемы, характеризующие базы данных с разных сторон с целью определить оптимальное построение информационной системы.

Ядро базы данных — внутренняя структура СУБД, обеспечивающая доступ ко всем компонентам базы данных. В новых версиях СУБД Access называется Microsoft Data Engine (MSDE); в ранних версиях ядро базы данных называлось *машина базы данных Microsoft Jet*. Ядро базы данных обеспечивает поддержку символов различных алфавитов, синтаксис языка SQL и другие средства обработки различных типов данных.

Пользователь БД — программа или человек, обращающийся к базе данных.

Запрос — процесс обращения пользователя к БД с целью ввести, получить или изменить информацию.

Транзакция — последовательность операций модификации данных в БД, переводящая ее из одного непротиворечивого состояния в другое непротиворечивое состояние.

Логическая структура БД — определение БД на физически независимом уровне, что ближе всего соответствует концептуальной ее модели.

Топология БД, или *структура распределенной БД*, — схема распределения физической организации базы данных в сети.

Локальная автономность — понятие, означающее, что информация локальной БД и связанные с ней определения данных принадлежат локальному владельцу и им управляют.

Удаленный запрос — запрос к базам данных, находящихся на ресурсах локальной сети предприятия или сети Интернет.

Возможность реализации удаленной транзакции — обработка одной транзакции, состоящей из множества SQL-запросов, на одном удаленном узле.

Поддержка распределенной транзакции — обработка транзакции, состоящей из нескольких SQL-запросов, выполняемых на нескольких узлах сети (удаленных или локальных), но каждый из которых обрабатывается только на одном узле.

Распределенный запрос — запрос, при обработке которого используются данные из БД, расположенные в разных узлах сети.

Системы распределенной обработки данных в основном связаны с первым поколением БД, которые строились на мультипрограммных операционных системах, хранились на устройствах внешней памяти центральной ЭВМ и использовали терминальный многопользовательский режим доступа. При этом пользовательские терминалы не имели собственных ресурсов, т. е. процессоров и памяти, которые могли бы использоваться для хранения и обработки данных. Первой полностью реляционной системой, работающей в многопользовательском режиме, была СУБД SYSTEM R фирмы IBM. Именно в ней были реализованы как язык манипулирования данными SQL, так и основные принципы синхронизации, применяемые при распределенной обработке данных, которые до сих пор являются базисными практически во всех коммерческих СУБД.

1.2. Архитектуры клиент—сервер в технологии управления удаленными базами данных

Вычислительная модель клиент-сервер исходно связана с появлением открытых систем в 1990-х гг. Термин *клиент—сервер* применялся к архитектуре программного обеспечения, состоящего из двух процессов обработки информации: клиентского и серверного. Клиентский процесс запрашивал некоторые услуги, а серверный — обеспечивал их выполнение. При этом предполагалось, что один серверный процесс может обслужить множество клиентских процессов. Учитывая, что аппаратная реализация этой моде-

ли управления базами данных связана с созданием локальных вычислительных сетей предприятия, такую организацию процесса обработки информации называют архитектурой клиент — сервер.

Основной принцип модели клиент — сервер применительно к технологии управления базами данных заключается в разделении функций стандартного интерактивного приложения на пять групп, имеющих различную природу:

- функции ввода и отображения данных (Presentation Logic);
- прикладные функции, определяющие основные алгоритмы решения задач приложения (Business Logic);
- функции обработки данных внутри приложения (DataBase Logic);
- функции управления информационными ресурсами (DataBase Manager System);
- служебные функции, играющие роль связок между функциями первых четырех групп.

Структура типового приложения, работающего с базой данных в архитектуре клиент — сервер, приведена рис. 1.1.

Как видно из рис. 1.1, клиентская часть приложения включает в себя следующие части:

- презентационную логику;
- бизнес-логику, или логику собственно приложений;
- логику обработки данных;
- процессор управления данными.

Презентационная логика (Presentation Logic) как часть приложения определяется тем, что пользователь видит на своем экране, что приложение работает. Сюда относятся все интерфейсные экранные формы, которые пользователь видит или заполняет в ходе работы приложения, а также все то, что выводится пользователю на экран в качестве результатов решения некоторых про-

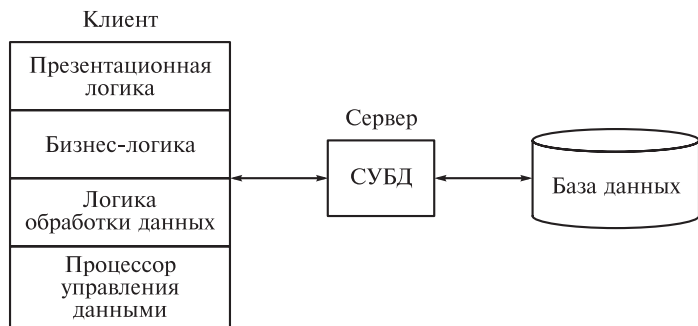


Рис. 1.1. Структура типового приложения, работающего с базой данных

межуточных задач либо как справочная информация. Следовательно, основными задачами презентационной логики являются:

- формирование экранных изображений;
- чтение и запись в экранные формы информации;
- управление экраном;
- обработка движений мыши и нажатие клавиш клавиатуры.

Бизнес-логика, или логика собственно приложений (Business Processing Logic), — это часть кода приложения, которая определяет собственно алгоритмы решения конкретных его задач. Обычно этот код записывается с использованием различных языков программирования, таких как С, С++, Visual Basic и др.

Логика обработки данных (Data Manipulation Logic) — это часть кода приложения, которая непосредственно связана с обработкой данных внутри него. Данными управляет собственно СУБД, а для обеспечения доступа к ним используется язык SQL.

Процессор управления данными (DataBase Manager System Processing) — это собственно СУБД, которая обеспечивает хранение и управление базами данных. В идеале функции СУБД должны быть скрыты от бизнес-логики приложения, однако при рассмотрении архитектуры приложения мы выделим их в отдельную его часть.

Модели распределений	Компоненты приложения						Пользователь
	Функции логики представления DP		Функции бизнес-логики		Функции управления данными		
			RP	RBL	RDM	DDM	
Распределенное представление (DP)							Клиент
							Сервер
Удаленное представление (RP)							Клиент
							Сервер
Распределенная бизнес-логика (RBL)							Клиент
							Сервер
Удаленное управление данными (RDM)							Клиент
							Сервер
Распределенное управление данным (DDM)							Клиент
							Сервер
Совмещение RBL и DDM							Клиент
							Сервер

Рис. 1.2. Распределение функций компонентов приложения в моделях клиент—сервер

В централизованной архитектуре (Host-Based Processing) указанные части приложения располагаются в единой среде и комбинируются внутри одной исполняемой программы. В децентрализованной архитектуре эти части приложения могут быть по-разному распределены между серверным и клиентским процессами.

В зависимости от характера распределений задач можно выделить следующие их модели (рис. 1.2):

- распределенное представление (Distribution Presentation);
- удаленное представление (Remote Presentation);
- распределенная бизнес-логика (Remote Business Logic);
- удаленное управление данными (Remote Data Management);
- распределенное управление данными (Distributed Data Management).

Эта условная классификации показывает, как могут быть распределены отдельные задачи между серверным и клиентскими процессами. В данной классификации отсутствует реализация удаленной бизнес-логики, так как считается, что она не может быть удалена полностью, а может быть лишь распределена между разными процессами, которые могут взаимодействовать друг с другом.

1.3. Двухуровневые модели

Двухуровневые модели управления БД фактически являются результатом распределения пяти указанных ранее групп функций стандартного интерактивного приложения между двумя процессами, выполняемыми на двух платформах: компьютере клиента и на сервере. В чистом виде не существует ни одна из них, однако рассмотрим наиболее характерные особенности каждой двухуровневой модели.

Модель удаленного управления данными, или модель файлового сервера (File Server — FS). В этой модели презентационная логика и бизнес-логика располагаются на клиентской части. На сервере располагаются файлы с данными и поддерживается доступ к этим файлам. Функции управления информационными ресурсами в этой модели находятся на клиентской части.

Распределение функций компонентов приложения в моделях клиент — сервер представлено на рис. 1.3.

В этой модели файлы базы данных хранятся на сервере, клиент обращается к серверу с файловыми командами, а механизм управления всеми информационными ресурсами — собственно база метаданных (выбранных данных) — находится на компьютере клиента.

Достоинство данной модели состоит в том, что приложение разделено на два взаимодействующих процесса. При этом сервер

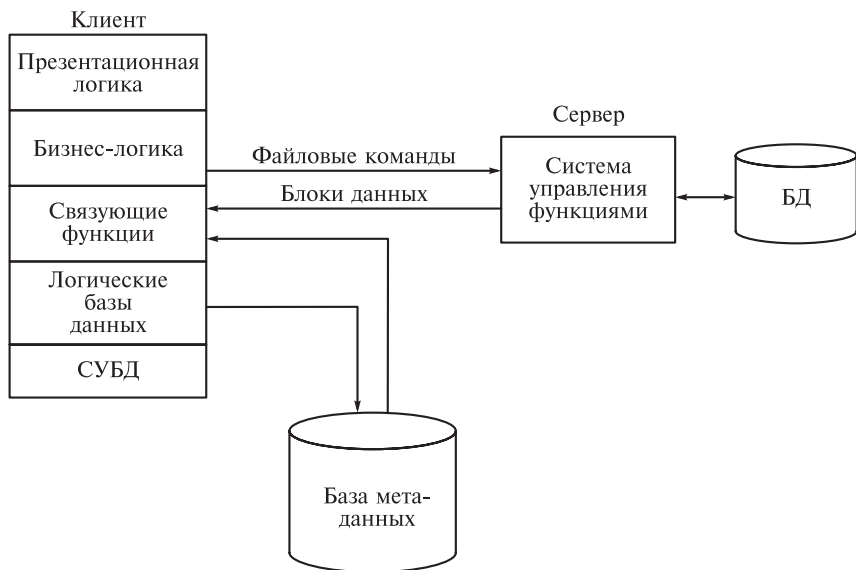


Рис. 1.3. Модель файлового сервера

(серверный процесс) может обслуживать множество клиентов, которые обращаются к нему с запросами. Собственно СУБД должна находиться в этой модели на компьютере клиента.

Алгоритм выполнения клиентского запроса сводится к следующему.

1. Запрос формулируется в командах языка манипулирования данными (ЯМД).
2. СУБД переводит этот запрос в последовательность файловых команд.
3. Каждая файловая команда вызывает перекачку блока информации на компьютер клиента, а СУБД анализирует полученную информацию, и если в полученном блоке не содержится ответ на запрос, принимается решение о перекачке следующего блока информации и т.д.
4. Перекачка информации с сервера на клиентский компьютер производится до тех пор, пока не будет получен ответ на запрос клиента.

Рассмотренная модель имеет следующие недостатки:

- высокий сетевой трафик, который связан с передачей по сети множества блоков и файлов, необходимых приложению;
- узкий спектр операций манипулирования с данными, определяемый только файловыми командами;
- отсутствие адекватных средств безопасности доступа к данным (защита только на уровне файловой системы).