

## Лекция 1. Введение в Oracle

### Семейство продуктов Oracle Database

Oracle Database 11g - последний представитель продуктов, составляющих семейство реляционных систем управления базой данных (СУБД) Oracle, построенных на основе единых исходных текстов. В это семейство входят:

#### *Oracle Enterprise Edition*

Флагманский продукт и основная тема настоящей книги. Ориентирован на крупномасштабные проекты, нуждающиеся в полном наборе средств Oracle. Только Enterprise Edition поддерживает такие развитые механизмы обеспечения безопасности, как виртуальная частная база данных (Virtual Private Database, VPD), детальный аудит (Fine-Grained Auditing) и другие опции, включая Database Vault, Advanced Security и Label Security. Лишь в Enterprise Edition хранилища данных поддерживают сжатие повторяющихся значений, кросс-платформенные переносимые табличные пространства, управление жизненным циклом информации (Information Lifecycle Management, ILM), перезапись запросов с материализованными представлениями, а также секционирование (Partitioning), OLAP и добычу данных (Data Mining). К числу механизмов обеспечения высокой доступности, включенных в Enterprise Edition, относятся Data Guard, ретроспективные (flashback) базы, ретроспективные таблицы и ретроспективные транзакции. В Oracle Database 11g добавлена опция сжатия Advanced Compression Option для любой рабочей нагрузки, в том числе для обработки транзакций, хранения больших объектов (Large Object, LOB) и резервных копий; подсистема тестирования базы данных, которая называется Real Application Testing Option и включает в себя программы Database Replay и SQL Performance Analyzer, а также опция Total Recall Option, обеспечивающая режим архивации ретроспективных данных Flashback Data Archive, который сохраняет данные, необходимые для выполнения хронологических запросов (запросов с конструкцией AS OF, где задается дата в прошлом).

#### *Oracle Standard Edition*

Эта СУБД ориентирована на реализацию баз данных малого и среднего размера. Ее можно развернуть в серверной конфигурации, имеющей до 4 ЦП, на одном компьютере или на кластере с использованием подсистемы Real Application Clusters (RAC).

#### *Oracle Standard Edition One*

Ориентированная на небольшие проекты, эта СУБД поддерживает до двух ЦП и не поддерживает RAC. В остальном набор возможностей схож с реализованным в редакции Oracle Standard Edition.

#### *Oracle Personal Edition*

СУБД, используемая разработчиками-одиночками для создания кода, который будет выполняться в многопользовательской СУБД.

В отличие от Express Edition, требует лицензии, но обладает всей функциональностью Enterprise Edition.

#### *Oracle Express Edition*

СУБД начального уровня, доступная для Windows и Linux бесплатно. Может использовать не более 1 Гбайт памяти и 4 Гбайт дискового пространства. Предоставляет часть функциональности, включенной в редакцию Standard Edition One. Отсутствуют такие функции, как виртуальная Java-машина, управляемое сервером резервное копирование и восстановление, а также подсистема Automatic Storage Management. Oracle Enterprise Manager не умеет управлять этой СУБД, однако ее можно развернуть так, что она будет доступна из административного интерфейса Oracle Application Express (бывший HTML-DB), позволяющего управлять несколькими пользователями.

Обычно Oracle выпускает новые версии своей флагманской СУБД каждые три-четыре года. Новые версии, как правило, посвящены какой-то одной теме и включают целый ряд новых функций. В последних версиях тема обозначалась в названии версии продукта. Так, в 1998 году Oracle анонсировала версию Oracle8i, где буква *i* обозначала поддержку развертывания для работы в Интернете. Версия Oracle9i продолжила эту тему. В 2003 году вышла версия Oracle Database 10g, где *g* означает сконцентрированность на моделях развертывания с поддержкой grid-вычислений. Oracle продолжает эту тему и в текущей версии СУБД, которая рассматривается в настоящей книге. Между основными версиями Oracle выпускает промежуточные. В них тоже добавляются новые возможности, но основное внимание все же уделено совершенствованию уже реализованных средств.

Термины «Oracle», «Oracle8», «Oracle8i», «Oracle9i», «Oracle Database 10g» и «Oracle Database

llg» в этой книге не всегда употребляются строго, поскольку Oracle Database llg включает все функции из предшествующих версий. Описывая функцию, которая впервые была включена в некую конкретную версию, во избежание путаницы мы старались отмечать этот факт, понимая, что многие читатели работают не с самой свежей версией Oracle. При описании функций, общих для всех этих версий, мы говорим просто «Oracle».

С 1983 года подразделение Oracle Development ведет разработку на основе модели единого набора исходных текстов для всего семейства продуктов, связанных с базами данных. Хотя в реализации каждой СУБД на самых нижних уровнях встречается системно-зависимый код, необходимый для лучшего учета особенностей конкретной платформы, интерфейсы, раскрываемые пользователям, разработчикам и администраторам, одинаковы. Поскольку поведение функций не зависит от платформы, любая организация может безболезненно перенести СУБД Oracle и приложения для них с одной аппаратной платформы или операционной системы на другую. Такая стратегия позволяет Oracle реализовывать новые функции только один раз для каждого набора продуктов.

## **Сводка функций СУБД Oracle**

СУБД Oracle - очень крупный продукт. Чтобы создать первичное представление о нем, мы начнем с высокоуровневого обзора основной функциональности.

Чтобы как-то структурировать широкий спектр возможностей СУБД Oracle, мы выделили следующие аспекты:

- средства разработки приложений базы данных;
- средства установления соединения с базой данных;
- распределенные базы данных;
- средства перемещения данных;
- средства повышения производительности;
- средства управления базой данных;
- средства обеспечения безопасности базы данных.

## **Средства разработки приложений баз данных**

СУБД Oracle обычно используется для хранения данных, которые извлекаются приложениями. Описанные в этом разделе средства и соответствующие продукты применяются для создания таких приложений. Мы решили отдельно рассмотреть программирование баз данных и возможности их расширения.

## **Программирование баз данных**

Во все варианты СУБД Oracle включены языки и интерфейсы, позволяющие программистам извлекать данные из базы и манипулировать ими. Средства программирования баз данных обычно интересуют разработчиков, которые создают коммерческие приложения на базе Oracle, а также ИТ-отделы, создающие приложения для нужд собственных организаций. Для доступа к данным в Oracle можно использовать SQL, ODBC, JDBC, SQLJ, OLE DB, ODP.NET, SQL/XML, XQuery и WebDAV. Программы, хранящиеся в самой базе данных, могут быть написаны на языках PL/SQL и Java.

### **SQL**

Описываемый стандартом ANSI язык Structured Query Language (SQL) включает базовые средства манипулирования данными, управления транзакциями и извлечения записей из базы данных. Бизнес-пользователи по большей части взаимодействуют с Oracle посредством приложений или инструментов бизнес-анализа, которые предоставляют интерфейсы, скрывающие SQL и присущую ему сложность.

### **PL/SQL**

PL/SQL- это разработанное Oracle процедурное расширение языка SQL. Обычно на нем реализуются логические программные модули для приложений. На языке PL/SQL можно писать хранимые процедуры, триггеры, циклы, условные предложения и обработку ошибок. Процедуры на PL/SQL можно откомпилировать и сохранить в базе данных. Блоки, написанные на PL/SQL, можно также исполнять непосредственно с помощью интерактивного инструмента SQL\*Plus, имеющегося во всех версиях Oracle. Программные блоки на PL/SQL можно скомпилировать заранее.

### **Java**

В Oracle8i язык Java впервые начал использоваться для написания хранимых процедур, а виртуальная Java-машина (JVM) была встроена непосредственно в СУБД (первоначальное

название JServer). JVM обеспечивает поддержку написания на Java хранимых процедур, методов и триггеров, а также технологий Enterprise JavaBeans™ (EJB), CORBA, POP и HTTP.

Включение Java в СУБД Oracle позволяет программистам, владеющим Java, применить свои знания к разработке приложений для Oracle. Java-приложения можно развертывать на стороне клиента, внутри сервера приложений или в базе данных - в зависимости от конкретных обстоятельств. Oracle Database 11g включает JIT-компилятор Java, по умолчанию активированный.

### **Oracle и веб-службы**

Начиная с версии Oracle Database 11g, СУБД может служить поставщиком веб-служб, реализованных в базе данных с помощью технологии XML DB. Веб-службы позволяют создавать запросы на языках SQL или XQuery и получать результаты в формате XML либо вызывать PL/SQL-функции или функции в составе пакета и получать их результаты. Реализация XQuery в Oracle Database 11g поддерживает пока еще обсуждаемый стандарт JSR-225 и включает ряд мер, повышающих производительность.

### **Большие объекты**

Интерес к применению больших объектов (LOB) постоянно растет, особенно в контексте хранения таких нетрадиционных типов данных, как изображения. В базе данных Oracle уже довольно давно можно было хранить большие объекты. В Oracle8 появилась возможность иметь в одной таблице несколько LOB-столбцов. В Oracle Database 10g по существу было снято ограничение на размеры больших объектов. В Oracle Database 11g внедрена технология SecureFiles, что заметно повысило производительность операций выборки и вставки больших объектов. Для LOB-данных с применением SecureFiles поддерживается прозрачное шифрование.

### **Объектно-ориентированное программирование**

Инфраструктура объектов для поддержки объектно-ориентированного подхода в программировании существовала со времен Oracle8. Например, программист мог создать определяемый пользователем тип данных, содержащий методы и атрибуты. Поддержка объектов в Oracle включает механизм Object Views, с помощью которого объектно-ориентированные программы могут работать с уже хранящимися в базе реляционными данными. Хранить объекты в базе данных можно в виде массивов переменной длины (VARRAY), вложенных таблиц или индекс-таблиц (index organized tables, IOT).

### **Языки третьего поколения (3GL)**

Программисты могут обращаться к базе данных Oracle из программ, написанных на языках C, C++, Java или COBOL, встраивая в них команды SQL. Перед тем как подавать такое приложение на вход платформенного компилятора, его необходимо пропустить через прекомпилятор. Последний заменяет команды SQL вызовами библиотечных функций, понятных стандартному компилятору. Oracle поддерживает такую методику с помощью дополнительного прекомпилятора Pro\*C для языков C и C++ и прекомпилятора Pro\*COBOL для языка COBOL. В последние версии Oracle включен прекомпилятор SQLJ для языка Java, который заменяет команды SQL обращения к библиотеке времени выполнения SQLJ, также написанной на Java.

### **Драйверы базы данных**

Во все версии Oracle включены драйверы, позволяющие приложению обращаться к базе данных посредством ODBC (открытый стандарт взаимодействия с базами данных) или JDBC (открытый стандарт взаимодействия с базами данных для Java). Имеются также поставщики данных для OLE-DB и .NET.

### **Интерфейс уровня вызовов Oracle**

Опытный программист, стремящийся добиться максимальной производительности, может определить команду SQL в виде символьной строки объемлющего языка, затем явно разобрать эту команду, привязать к ней переменные и выполнить ее с помощью интерфейса уровня вызовов Oracle (Oracle Call Interface, OCI). Интерфейс OCI гораздо детальнее предыдущих, для работы с ним и последующей отладки программисту придется затратить много времени и усилий. Разработка приложений с помощью OCI может занять много времени, но расширение функциональности и повышение быстродействия оправдают дополнительные затраты. Например, если механизм обеспечения высокой доступности реализован так, что несколько систем разделяют общие диски с помощью подсистемы Real Application Clusters, то OCI дает возможность написать программу, которая позволит пользователю прозрачно присоединиться

ко второму серверу, если первый выйдет из строя.

## **Поддержка национальных языков**

Подсистема поддержки национальных языков (National Language Support, NLS) предоставляет наборы символов и прочие данные, например форматы записи чисел и дат, для различных языков. В Oracle Database 11g добавлена поддержка Unicode5.0. Кодировка Unicode позволяет хранить все данные или постепенно переводить на нее отдельные столбцы. Кодировки UTF-8 и UTF-16 обеспечивают поддержку более 57 языков и 200 наборов символов. Многие вещи локализованы изначально (например, форматы данных), но при желании с помощью утилиты Oracle Locale Builder можно создать нестандартную локаль. Включен также инструментальный Globalization Toolkit для создания приложений, поддерживающих несколько языков.

## **Расширяемость базы данных**

Работа в Интернете и в корпоративных сетях интранет выдвигает новые требования к хранению данных нетрадиционных типов и манипулированию ими. Если нужно расширить стандартную функциональность базы данных для хранения изображений, аудио, видео, пространственных данных и временных рядов, то эти возможности можно добавить путем расширения стандартного языка SQL.

## **Подсистема Oracle Multimedia**

Подсистема Oracle Multimedia (бывшая mterMedia) предоставляет средства манипулирования текстом, изображениями, аудио- и видеоинформацией, географическими координатами, а именно:

- часть Multimedia, относящаяся к тексту (Oracle Text), может распознать смысл документа, производя в нем поиск по темам и ключевым фразам;
- часть Multimedia, относящаяся к изображениям, умеет сохранять и извлекать изображения в различных форматах; начиная с версии Oracle Database 11g, поддерживается формат DICOM медицинских изображений;
- части Multimedia, относящиеся к аудио- и видеоинформации, способны сохранять и извлекать аудио- и видеоклипы соответственно;
- часть Multimedia, относящаяся к геоинформации, умеет извлекать данные о пространственных координатах.

## **Управление контентом в Oracle**

К средствам управления контентом относится подсистема Content Database Option, позволяющая сохранять в базе данных документы, а также приложения для управления контентом компании Stellent, приобретенной Oracle в 2007 году: Universal Content Management, Universal Records Management и Information Rights Management.

## **Средства поиска в Oracle**

В состав продуктов Oracle Database и Application Server входит инструмент поиска Ultra Search. Обычно он применяется для сбора информации о местонахождении различных текстовых данных, хранящихся в корпоративной сети. Выборка документов базируется на правах доступа конкретного пользователя. Кроме того, предлагается альтернативная система Secure Enterprise Search, обладающая большей гибкостью в среде, не основанной целиком на продуктах Oracle.

## **Подсистема Oracle Spatial Option**

Подсистема Oracle Spatial Option включена только в редакцию Oracle Enterprise Edition. Она позволяет оптимизировать выборку и отображение данных, привязанных к координатам, и применяется при разработке геоинформационных систем (ГИС). Некоторые производители таких систем уже включили ее в свои продукты и применяют в качестве механизма поиска и выборки.

## **XMLDB**

Поддержка типа данных XML была встроена в СУБД Oracle9i. Структурированный XML-объект хранится в объектно-реляционной базе данных в соответствии со спецификацией W3C DOM. Встраивание синтаксиса XPath в поисковые запросы на языке SQL отвечает спецификациям группы SQLX. Язык XQuery также поддерживается.

## **Средства установления соединения с базой данных**

Установление соединения между клиентом и сервером базы данных - ключевой компонент всей архитектуры. По этому соединению передаются все данные, запрашиваемые приложением. В Oracle включены различные средства для установления и настройки соединения с базой данных

(описаны в отдельных разделах ниже). Мы разбили обсуждение на две части: сетевые компоненты СУБД и продукт Oracle Application Server.

## Сетевые компоненты СУБД

Пользователи подключаются к базе данных, устанавливая с ней соединение по сети. Можно также связать между собой по сети различные серверы базы данных. Oracle предлагает несколько способов установления соединений между пользователем и базой данных или между различными серверами баз данных, как описано в следующих разделах.

### Oracle Net

Интерфейс с сетью Oracle Net в версии Oracle8 назывался Net8, а в более ранних версиях - SQL\*Net. Он поддерживает широкий спектр сетевых протоколов, хотя самый распространенный сегодня - TCP/IP. Средства, ассоциируемые с Oracle Net, например разделяемые серверы, в совокупности называются Oracle Net Services.

### Oracle Internet Directory

Служба интернет-каталогов Oracle Internet Directory (OID) впервые появилась в версии Oracle8L. OID заменила прежнюю службу Oracle Names, поскольку позволяет пользователю соединиться с сервером Oracle Server, не создавая конфигурационный файл на стороне клиента. OID представляет собой LDAP-совместимый каталог (Lightweight Directory Access Protocol), а потому поддерживает Oracle Net и другие протоколы на основе LDAP.

### Oracle Connection Manager

Каждое соединение с базой данных потребляет дефицитные сетевые ресурсы, и это может отразиться на производительности приложения. Менеджер соединений (Connection Manager, CMAN), показанный на рис. 1.3, позволяет уменьшить количество сетевых соединений клиентов Oracle Net с сервером за счет применения *концентраторов*, задача которых - мультиплексировать соединения, объединив несколько логических соединений в одно физическое. Достоинства механизма мультиплексирования соединений становятся очевидными при большом количестве активных пользователей.



Рис. 1.1. Концентраторы и менеджеры соединений при большом количестве пользователей

Менеджер соединений позволяет также работать с несколькими сетевыми протоколами, если в сети имеются клиенты или серверы, не использующие TCP/IP. В версии Oracle Database 10# появилась возможность динамически конфигурировать менеджер соединений, то есть изменять его параметры, не останавливая процесс CMAN.

## Oracle Application Server

Широкое распространение приложений для Интернета и сетей интранет стало причиной перехода от архитектуры клиент/сервер (когда значительные части приложения реализованы в виде «толстых» клиентов) к трехуровневой архитектуре (когда браузер предоставляет все, что нужно «тонкому» клиенту). Сервер приложений Oracle Application Server позволяет развернуть промежуточный слой трехуровневой архитектуры для веб-приложений, компонентных

приложений и интеграции приложений масштаба предприятия. Oracle Application Server - основная часть продукта Fusion Middleware, допускающая масштабирование на несколько серверов промежуточного слоя.

Этот продукт включает веб-прослушиватель на базе популярного сервера Apache, сервлеты и сценарии JavaServer Pages (JSP), бизнес-логику и/или компоненты для доступа к данным. Бизнес-логика часто разворачивается в виде компонентов Enterprise JavaBeans (EJB). Компоненты для доступа к данным могут быть написаны с применением JDBC, SQLJ и EJB. TopLink - это инструмент отображения, который связывает Java-объекты с базой данных через JDBC, так что разработчик на Java может не думать о конструировании вызовов SQL и об ошибках приложения, вызванных изменениями в схеме базы данных. Oracle Application Server предлагает также механизм кэширования и готовые решения задач, возникающих при создании порталов, систем бизнес-анализа и беспроводного доступа.

#### *Кэширование*

Компонент Oracle Application Server Web Cache реализует промежуточный уровень для кэширования веб-страниц целиком или частично. Предшествующий механизм Oracle Application Server Database Cache, который использовался для кэширования PL/SQL-процедур и анонимных PL/SQL-блоков, начиная с версии Oracle Application Server 10g не поддерживается.

#### *Портал*

Компонент Oracle Application Server Portal входит также в продукт Oracle Developer Suite и применяется для создания простых в использовании корпоративных порталов. Разработанный портал разворачивается внутри Application Server.

#### *Бизнес-анализ*

В состав продукта Application Server Business Intelligence входит компонент Portal, а также оригинальные инструменты бизнес-анализа, разработанные Oracle:

- Oracle Reports - масштабируемый промежуточный слой для вывода результатов заранее заданных запросов в виде отчетов;
- Oracle Discoverer для предъявления произвольных запросов и анализа результатов;
- платформа разворачивания для разработанных в JDeveloper приложений для OLAP-обработки и добычи данных.

#### *Oracle Wireless*

В состав компонента Oracle Wireless (бывший Oracle Portal-to-Go) входят:

- контент-адаптеры для преобразования информационного содержимого в формат XML;
- преобразователи форматов (device transformer) для преобразования из XML в язык разметки, поддерживаемый конкретным устройством;
- порталы персонализации для персонализации оповещений, адресов назначения оповещений, адресных меток (location mark) и профилей; кроме того, беспроводной портал персонализации применяется для создания, обслуживания, тестирования и публикации URL службы, а также для управления пользователями.

Продукт Oracle Application Server поставляется в нескольких редакциях: Enterprise Edition, Standard Edition, Standard Edition One и Java Edition; последний включает компоненты, необходимые разработчикам на Java. В Standard Edition и Standard Edition One включены компоненты Portal, TopLink вместе с Application Development Framework и Web Cache. В Enterprise Edition добавлены следующие компоненты: Forms Services, Reports Services, Discoverer Viewer, Oracle Internet Directory, Oracle Application Interconnect, Wireless Option и интеграция с Enterprise Service Bus (ESB). В Java Edition входят компоненты HTTP Server, OC4J и TopLink вместе с Application Development Framework.

Для редакции Oracle Application Server Enterprise Edition имеется еще ряд дополнительных опций:

#### *BPEL Process Manager Option*

Инструмент Business Process Execution Language (BPEL, язык исполнения бизнес-процессов) спроектирован для работы в сервисноориентированных архитектурах (SOA) и применяется для создания, администрирования и разворачивания бизнес-процессов, связывающих несколько приложений. Он поддерживает стандарты BPEL, Web Services, XML, XSLT, XPath, JMS и JCA.

#### *BusinessActivity Monitoring (BAM)*

Компонент BAM служит для построения инструментальных панелей реального времени, на которых отображаются основные индикаторы производительности (key performance indicator, KPI), содержащие данные от оповещений, поступающих через Сеть.

### *BI Publisher*

Инструмент форматирования отчетов, применяемый для генерирования высококачественных отчетов на основе данных в формате XML.

### *Service Registry*

Реестр служб Oracle Service Registry позволяет публиковать информацию о службах и ссылку на авторитетную систему (System of Record) для SOA-служб.

### *Комплект SOA Suite для Oracle Middleware*

В этот комплект входят компоненты Oracle Fusion Middleware для SOA: BPEL, BAM, движок бизнес-правил, Enterprise Service Bus (механизм обмена сообщениями, маршрутизации и трансформации), Web Services Management (включает менеджер политик и инструментальную панель мониторинга), Web Services Registry, а также адаптеры приложений и технологий.

### *Communication and Mobility Server*

В этот продукт входит компонент TimesTen, а также SIP Servlet Container, каркас активации и активаторы, средства голосового и мобильного доступа.

### *WebCenter*

WebCenter - последняя разработанная Oracle инфраструктура для построения порталов. Применяется для развертывания портлетов Ajax-компонентов, особенно для приложений, следующих принципам Web 2.0. Включает форумы, сервер присутствия, клиент системы мгновенной передачи сообщений, Wiki, установление и разрыв VOIP-вызова, SIP Servlet Container, API для Java и веб-служб, интеграцию с системой Click-2-dial и программный клиент с поддержкой голосовой связи.

### *Адаптеры для Fusion Middleware*

Имеются адаптеры для приложений, мониторов обработки транзакций, EDI и другие.

Комплект Fusion Middleware SOA Suite служит основой архитектуры интеграции приложений Application Integration Architecture (AIA). B2B включены также готовые бизнес-объекты и бизнес-процессы под общим названием Process Integration Packs. Эта архитектура является фундаментом для интеграции существующих и будущих приложений Oracle.

## **Распределенные базы данных**

СУБД Oracle славится умением обрабатывать очень большие объемы данных и поддерживать множество одновременно работающих пользователей. Oracle не только хорошо масштабируется для развертывания на все более мощных одиночных системах, но может быть развернута и в распределенной конфигурации. Экземпляры Oracle, развернутые на нескольких платформах, можно объединить, представив в виде логически единой распределенной базы данных.

В этом разделе описаны основные способы взаимодействия между базами данных в распределенной системе.

## **Распределенные запросы и транзакции**

Корпоративные данные часто распределены по нескольким базам из соображений емкости и распределения сфер ответственности. Но пользователям бывает нужно запрашивать или обновлять распределенные данные так, как будто они находятся в одной базе.

Корпорация Oracle первой ввела распределенные базы данных еще в начале 1980-х в ответ на требования организовать доступ к данным на разных платформах. *Распределенные запросы* позволяют извлекать данные из нескольких баз. *Распределенные транзакции* служат для вставки, удаления или обновления данных, находящихся в распределенной базе. Механизм двухфазной фиксации гарантирует, что все серверы баз данных, участвующие в транзакции, либо зафиксируют, либо откатят ее. Фоновые процессы восстановления гарантируют непротиворечивость базы данных при сбое системы во время обработки распределенной транзакции. Когда отказавшая система станет доступна, тот же самый процесс завершит распределенные транзакции.

Распределенные транзакции можно реализовать также с помощью распространенных мониторов транзакций (TP), которые взаимодействуют с Oracle по стандартному (X/Open) интерфейсу XA. В Oracle8i был добавлен механизм координации транзакций посредством сервера Microsoft Transaction Server (MTS), поэтому теперь распределенная транзакция, инициированная MTS, может распространяться и на базу данных Oracle.

## **Heterogeneous Services**

Компонент Heterogeneous Services позволяет обращаться из СУБД Oracle к данным, хранящимся в других базах, и сторонним службам с помощью обобщенных интерфейсов

установления связи ODBC и OLE-DB.

Дополнительный компонент Transparent Gateways пользуется агентами, специально разработанными для различных оконечных систем. Этот компонент позволяет формулировать запросы на диалекте языка SQL для Oracle и отправлять их другой СУБД. При этом запрос автоматически и прозрачно для пользователя будет транслирован на диалект SQL, понятный источнику данных. Помимо предоставления доступа к сторонним SQL-службам компонент Heterogeneous Services реализует транзакционность с помощью протокола двухфазной фиксации Oracle для других баз данных и процедурных служб, которые вызывают написанные на языке третьего поколения функции в системах, не управляемых Oracle. Пользователь взаимодействует с базой данных Oracle так, будто все объекты хранятся в ней, а компонент Heterogeneous Services прозрачно обращается к «чужой» базе данных от имени пользователя.

## **Средства перемещения данных**

При использовании распределенных баз данных часто требуется перемещать данные из одной базы данных Oracle в другую. А иногда необходимо организовать несколько копий одной и той же базы в разных местах, чтобы уменьшить объем сетевого трафика или повысить доступность данных. Вы можете экспортировать сами данные и словари данных (метаданные) из одной базы и импортировать их в другую. В Oracle Database 10g для экспорта/импорта была реализована высокоскоростная помпа данных (data pump).

Oracle предлагает много других дополнительных средств этой категории: переносимые табличные пространства, компоненты Advanced Queuing и Oracle Streams, а также решения для извлечения, трансформации и загрузки (ETL) данных. Рассмотрим их подробнее.

## **Переносимые табличные пространства**

Переносимые табличные пространства впервые появились в версии Oracle8i. Вместо того чтобы запускать процесс экспорта/импорта, который сбрасывает данные и описывающие их структуры в промежуточный файл для последующей загрузки, можно перевести табличное пространство в режим чтения, перенести или скопировать его из одной базы в другую, а затем смонтировать. При этом в исходной и конечной базах словари, описывающие табличное пространство, должны быть одинаковыми. Такой метод позволяет сэкономить немало времени в случае перемещения больших объемов данных. Начиная с версии Oracle Database 10g можно переносить табличные пространства между различными платформами или операционными системами.

## **Advanced Queuing и Oracle Streams**

Компонент Advanced Queuing (AQ), впервые появившийся в версии Oracle8i, позволяет асинхронно посылать сообщения из одной базы данных Oracle в другую. Поскольку сообщения хранятся в очереди внутри базы данных и посылаются асинхронно, когда устанавливается соединение, накладные расходы и объем сетевого трафика оказываются гораздо ниже, чем при использовании традиционных способов гарантированной доставки с помощью протокола двухфазной фиксации транзакции, включающей исходную и конечную базы данных. Сохраняя же сообщения в базе, AQ обеспечивает более надежный механизм восстановления, чем при других реализациях очередей с хранением сообщений в файловой системе.

Наличие механизма передачи сообщений в Oracle открывает возможность разработки и развертывания решений на базе публикации/подписки с применением правил для определения подписавшихся приложений. Когда в списке рассылки публикуется новый контент, анализируются заданные для этого списка правила и принимается решение, каким подписчикам он должен быть отправлен. При таком подходе единственный список рассылки может эффективно обслужить потребности различных сообществ подписчиков. В первой версии Oracle9i в компонент AQ были добавлены поддержка XML и интеграция с каталогом Oracle Internet Directory (OID).

Во второй версии Oracle9i компонент AQ стал частью подсистемы Oracle Streams. Последняя состоит из трех основных компонентов: репликация по журналу для сбора данных, очереди для промежуточного хранения данных и определяемые пользователем правила потребления данных. Начиная с Oracle Database 10g Streams включает поддержку технологии Change Data Capture (отслеживание изменений в источниках данных) и передачи файлов. Подсистема Streams управляется из программы Oracle Enterprise Manager.

## **Извлечение, трансформация и загрузка данных**

Инструмент Oracle Warehouse Builder (OWB) служит для проектирования целевых баз данных, особенно используемых в качестве хранилищ (data warehouses), и предоставляет репозиторий метаданных. Однако он более широко известен как графический инструмент построения отображения исходной базы на конечную и генерации сценариев извлечения, трансформации



и загрузки данных (ETL). OWB пользуется средствами ETL, которые впервые были встроены в СУБД в версии Oracle9i. OWB поставляется в составе СУБД Oracle начиная с версии Oracle Database 10g Release 2.

Дополнительно Oracle предлагает инструмент интеграции данных Oracle Data Integrator (ODI), который не так тесно связан с СУБД Oracle, как OWB (хотя база данных Oracle может быть как исходной, так и конечной). Oracle Data Integrator основан на продукте компании Sunopsis, приобретенной Oracle. Помимо средств ETL ODI может генерировать код веб-служб для развертывания в архитектуре SOA и является ключевым компонентом стратегии интеграции с SOA, реализованной в Oracle.

## Средства повышения производительности

В Oracle имеется несколько механизмов, специально предназначенных для повышения производительности в определенных ситуациях. Мы отнесли их к двум категориям: распараллеливание работы базы данных и организация хранилищ данных.

### Распараллеливание работы базы данных

Распараллеливание повышает скорость выполнения запросов, настройки и обслуживания базы данных. Разбив одну задачу на несколько меньших подзадач, каждая из которых выполняется в отдельном процессе, можно весьма заметно повысить производительность некоторых операций в базе данных. Вот некоторые типы запросов, которые могут быть распараллелены:

- сканирование таблицы;
- вложенные циклы;
- соединение таблиц методом сортировки и слияния;
- группировка GROUP BY;
- подзапросы типа NOT IN (антисоединение);
- определенные пользователем функции;
- сканирование индекса;
- SELECT DISTINCT UNION и UNION ALL;
- соединение таблиц методом хеширования;
- ORDER BY и агрегирование;
- соединение типа «звезда» по битовым индексам (bitmap star joins);
- соединение по секциям (partition-wise join);
- хранимые процедуры (на языках PL/SQL и Java, а также внешние подпрограммы).

Помимо запросов распараллеливанию поддаются многие другие средства Oracle.

### Организация хранилищ данных и бизнес-анализ

Хотя распараллеливание повышает производительность СУБД Oracle в целом, к быстродействию систем бизнес-анализа и хранилищ данных предъявляются особые требования.

#### Битовые индексы

В Oracle 7.3 была добавлена поддержка битовых индексов, обеспечивающих быструю выборку некоторых типов данных. Лучше всего битовые индексы работают для столбцов, в которых число различных значений мало по сравнению с общим числом строк в таблице.

В битовом индексе не хранятся фактические значения. Вместо этого каждому возможному значению сопоставляется один бит, который равен 1, если строка содержит это значение, и 0 в противном случае.

#### Оптимизация запросов типа «звезда»

Типичный запрос к хранилищу данных адресован большой *таблице фактов*, которая связана внешними ключами с гораздо меньшими по размеру *таблицами измерений*. В версии Oracle 7.3 была реализована оптимизация таких *запросов типа «звезда»*. Выигрыш в производительности достигается за счет построения декартова произведения таблиц измерений и последующего единственного соединения с таблицей фактов. В Oracle8 этот механизм получил дальнейшее развитие и называется *параллельным соединением типа «звезда» по битовым индексам* (parallel bitmap star join). В нем используются битовые индексы по внешним ключам таблиц измерений, чтобы ускорить вычисление соединения типа «звезда» для большого количества таблиц измерений.

#### Материализованные представления

Начиная с Oracle8i, материализованные представления были еще одним способом существенно

повысить скорость выполнения запросов. Идея заключается в том, что информация, извлеченная из таблицы фактов, группируется по значениям полей из таблиц измерений, и полученные сводные данные сохраняются в виде материализованного представления. Если запрос может использовать это представление, то он прозрачно для пользователя переадресуется к нему. В каждой версии Oracle появляются новые способы оптимизации работы с материализованными представлениями.

### **Аналитические функции**

В Oracle и других СУБД все заметнее тенденция включать аналитические и статистические функции, доступные из SQL. Впервые такая возможность появилась в версии Oracle8i, когда были включены функции CUBE и ROLLUP. На сегодняшний день имеются также функции ранжирования, оконные агрегатные функции, функции запаздывания и опережения, линейная регрессия, дескриптивные статистики, корреляция, кросс-табуляция, проверка гипотез, подбор распределения и анализ Парето.

### **Подсистема OLAP Option**

Подсистема OLAP Option физически сохраняет многомерные кубы в реляционной базе Oracle. Чаще всего к этим кубам обращаются с помощью SQL, хотя имеется и Java API. Начиная с версии Oracle Database 11g оптимизатор Oracle распознает уровни внутри кубов. В результате любой инструмент бизнес-анализа может прозрачно для пользователя получить выигрыш от повышения производительности. Обновления значений в кубах теперь выполняются аналогично обновлению материализованных представлений.

### **Подсистема Data Mining Option**

Начиная с версии Oracle9i в СУБД встроены популярные алгоритмы добычи данных. Они включены в подсистему Data Mining Option, а обратиться к ним можно посредством PL/SQL или специального Java API. Приложения для добычи данных, в которых применяются эти алгоритмы, обычно пишутся с помощью программы DataMiner производства Oracle или инструментов, поставляемых компаниями-партнерами Oracle, например InforSense или SPSS. В подсистеме Data Mining Option для Oracle Database 11g реализованы следующие алгоритмы: наивная байесовская фильтрация, ассоциации, адаптивные байесовские сети, кластеризация, машины опорных векторов (SVM), факторизация неотрицательной матрицы (NMF), деревья решений и обобщенные линейные модели.

### **Инструменты бизнес-анализа**

К хранилищам данных в Oracle обычно обращаются из инструментов бизнес-анализа известных сторонних поставщиков. Однако по мере того как корпорация Oracle расширяет спектр предложений, приобретая другие компании, все чаще применяются предлагаемые ею программы. Изначально в состав Oracle включались только инструменты Oracle Discoverer и Reports (они до сих пор входят в состав Application Server или поставляются в составе отдельного продукта Oracle Business Intelligence Standard Edition Suite).

Флагманским продуктом Oracle в этой области является Oracle Business Intelligence Enterprise Edition Suite (OBI EE), который первоначально включал продукты бывшей компании Siebel Analytics: Oracle Answers, Dashboards, Delivers, BI Publisher и Office Plug-ins. В продукте OBI EE Plus это предложение было расширено за счет компонентов компании Hyperion: Foundation Services, Interactive Reporting, SQR production reporting, Financial Reporting, SmartView for Office и Web Analysis.

Для построения OLAP-куба и сопутствующих функций независимо от хранилища данных в базе теперь доступен компонент Essbase. Подмножество OBI EE включено в продукт Business Intelligence Standard Edition One вместе с редакцией СУБД Oracle Standard Edition One и Oracle Warehouse Builder.

Oracle предлагает также приложения бизнес-анализа, включающие средства моделирования данных, их анализа и генерации отчетов; при этом они уже заполнены готовыми метаданными о бизнесе. К флагманским приложениям относятся Oracle Business Intelligence Applications (прежнее название Siebel Business Analytics Applications) и Hyperion Financial Performance Management Applications.

### **Средства управления базой данных**

В Oracle включено много функций, упрощающих администрирование базы данных. Кардинально эта область улучшилась в версии Oracle Database 10g, а в Oracle Database 11g ее эволюция продолжилась в сторону большей автоматизации настройки и управления. Если вы продолжаете администрировать базы данных Oracle по старинке (например, с помощью сценариев), но собираетесь перейти на новую версию, то самое время пересмотреть свой подход.

Начиная с версии Oracle Database 10g статистика собирается автоматически и сохраняется в репозитории рабочей нагрузки Automatic Workload Repository (AWR) внутри базы данных.

Автоматический диагностический монитор базы данных Automatic Database Diagnostic Monitor (ADDM) периодически обрабатывает статистику и посылает оповещения о возможных проблемах программе Oracle Enterprise Manager, в которой можно проанализировать ситуацию более подробно и, если необходимо, принять меры. Некоторые функции, теперь полностью автоматизированные, например, Automatic Memory Management (автоматическое управление памятью), также пользуются данными, сохраняемыми в AWR.

Автоматизированные рекомендации, которые дает Oracle, основаны на состоянии базы данных, близком к реальному времени. Часто рекомендации оказываются более точными, чем было возможно раньше при использовании ручных процедур. В следующих разделах мы покажем, как это влияет на программу Oracle Enterprise Manager, дополнительные пакеты, подсистему Information Lifecycle Management, резервное копирование и восстановление, а также на доступность базы данных.

## **Oracle Enterprise Manager**

Программа Oracle Enterprise Manager (ЕМ) включена в большинство редакций СУБД. ЕМ предоставляет инфраструктуру для создания инструментов администрирования базы данных и HTML-интерфейс для управления пользователями, экземплярами и различными подсистемами. С помощью ЕМ можно также администрировать Oracle Application Server, Oracle Applications, операционную систему Linux и программные продукты других поставщиков.

На консоль базы данных в текущей версии Oracle выводится информация о состоянии базы данных, доступности, схеме, конфигурации средств перемещения данных и сопровождении ПО. В Oracle Database 11g появился новый инструмент Support Workbench со своей инфраструктурой диагностики, предназначенной для передачи информации о возникших проблемах в службу технической поддержки Oracle. Одновременно к репозиторию ЕМ могут обращаться несколько администраторов баз данных.

Развернуть ЕМ можно разными способами: как центральную консоль для мониторинга нескольких баз данных с помощью агентов, как «консоль продукта» (по умолчанию устанавливается вместе с каждой базой данных) или для удаленного доступа (этот режим иногда называют «студийным»). Если Enterprise Manager развернут как центральная консоль, то его называют «Центром управления решеткой» (Grid Control) и используют для быстрой установки программного обеспечения Oracle, подготовки к работе и автоматизированного наложения заплат.

Подмножество функциональности Enterprise Manager доступно в броузере Microsoft Pocket PC Internet Explorer для КПК с помощью программы EM2Go, способной следить за состоянием СУБД Oracle и сервера Oracle Application Server.

## **Information Lifecycle Management и ILM Assistant**

Подсистема управления жизненным циклом информации Information Lifecycle Management (ILM), появившаяся в 2006 году, предоставляет средства для определения классов данных и уровней хранения. При этом она перемещает данные на те уровни хранения, которые обеспечивают оптимальное сочетание производительности и стоимости. Интерфейс ILM Assistant для настройки и администрирования ILM можно загрузить с сайта Oracle Technology Network по адресу <http://otn.oracle.com>.

## **Резервное копирование и восстановление**

Каждому администратору базы данных известно, что резервное копирование - скучное, но необходимое занятие. Если резервная копия снята неправильно, то восстановление с нее сильно затруднено, а то и вовсе невозможно. К сожалению, важность этой повседневной задачи часто осознают лишь после утраты критически важных данных в результате сбоя системы. В следующих разделах мы расскажем о некоторых средствах, применяемых для резервного копирования.

### **Recovery Manager**

Основные виды резервных копий: полная копия базы данных (наиболее часто встречается), копия табличного пространства, копия файла данных, копия управляющего файла и копия архивного журнала. В версии Oracle8 появилась программа Recovery Manager (RMAN), с помощью которой сервер управляет резервным копированием и восстановлением базы данных, используя хранящийся в ней каталог восстановления (Recovery Catalog). RMAN умеет автоматически находить, копировать и восстанавливать (полностью или до определенного момента) файлы данных, управляющие файлы и архивные журналы. Начиная с версии Oracle9i RMAN может перезапускать процесс резервного копирования или восстановления и реализует политики окна восстановления по истечении срока хранения резервной копии. В Oracle

Enterprise Manager имеется графический интерфейс к RMAN. В версии Oracle Enterprise Manager 10g появился усовершенствованный планировщик заданий, с помощью которого можно настроить RMAN для автоматического запуска резервного копирования с записью на диск.

### **Инкрементное резервное копирование и восстановление**

В версии Enterprise Edition RMAN может также снимать инкрементные резервные копии баз данных. В этом случае копируются только блоки, модифицированные с момента снятия последней копии файла данных, табличного пространства или базы данных. Поэтому инкрементные копии получаются меньше, а восстановление с них выполняется быстрее, чем с полных. RMAN также умеет выполнять восстановление до заданного момента времени, что позволяет получить состояние данных непосредственно перед нежелательным событием (например, перед случайным удалением таблицы).

### **Oracle Secure Backup**

Программу RMAN применяют различные поставщики ПО для управления носителями, но начиная с версии Oracle Database 10g, в СУБД уже входит упрощенное решение для управления хранением резервных копий на магнитных лентах, которое называется Oracle Secure Backup XE. Дополнительно Oracle предлагает полномасштабное решение - Oracle Secure Backup.

### **Доступность базы данных**

Доступность базы данных зависит от надежности и правильности администрирования СУБД, операционной системы и аппаратных компонентов. Oracle повышает доступность за счет сокращения времени резервного копирования и восстановления. Достигается это следующими методами:

- возможность оперативного и параллельного резервного копирования и восстановления;
- улучшенное управление оперативными данными за счет секционирования;
- использование аппаратных средств для улучшенного мониторинга и перехвата управления при отказе.

Эти методы описаны в следующих разделах.

### **Секционирование**

Секционирование (partitioning) впервые введено в версии Oracle8 с целью повысить управляемость и доступность. Отдельные секции можно вывести из оперативного режима для обслуживания, сохранив доступ к остальным. При реализации хранилищ данных секционирование иногда применяется для организации скользящих окон, основанных на диапазонах дат. Имеются и другие варианты секционирования: хеш-секционирование (когда данные разносятся по секциям на основе значения хеш-функции, обеспечивающей равномерное распределение) и секционирование по списку значений ключа (данные распределяются по секциям исходя из дискретных значений, например, географического местоположения). Начиная с версии Oracle Database 11g можно также организовывать интервальное секционирование, в этом случае новые диапазоны создаются автоматически по мере вставки записей.

Различные виды секционирования можно сочетать для создания «составных» секций. Примерами могут служить секции типа «диапазон-диапазон», «диапазон-хеш», «диапазон-список», «список-диапазон», «список-хеш» и «список-список».

### **Data Guard**

Понятие резервной базы данных (standby database) впервые появилось в версии Oracle 7.3. Это копия рабочей базы данных, которая начинает использоваться, если последняя недоступна, например из-за сбоя основного сервера или во время профилактического обслуживания. Рабочая и резервная базы данных могут быть географически разнесены. Резервная база данных создается как копия рабочей и обновляется путем накатывания архивных журналов, создаваемых в процессе эксплуатации рабочей базы. Компонент Data Guard, появившийся в версии Oracle9i, полностью автоматизирует этот процесс; раньше копировать и накатывать журналы приходилось вручную. Агенты размещаются в местах расположения рабочей и резервной баз, а Data Guard Broker координирует выполнение команд. Единственная команда Data Guard инициирует восемь шагов, необходимых для перехвата управления при отказе.

Помимо поддержки физической резервной базы данных Data Guard может создавать логическую резервную базу. В этом случае архивные журналы Oracle преобразуются в транзакции SQL и накатываются на открытую резервную базу данных.

В Oracle Database 10g появилось несколько новых функций, в том числе поддержка наката данных из журнала в реальном времени, интеграция с ретроспективной (Flashback) базой данных и сжатие архивного журнала. Начиная с Oracle Database 10g поддерживается

пошаговое обновление (rolling upgrade). В версии Oracle Database 11g опция Active Data Guard Option позволяет выполнять в резервной базе данных запросы, сортировку и генерацию отчетов даже в то время, когда накатываются изменения из рабочей базы.

### **Fail Safe**

Подсистема Fail Safe повышает надежность базы данных Oracle. Перехват управления при отказе (failover) реализуется с помощью второй системы или узла, который обеспечивает доступ к данным, находящимся на разделяемом диске, в ситуации, когда первая система или узел выходит из строя. Подсистема Fail Safe для Windows в сочетании со службой Microsoft Cluster Services гарантирует перехват управления при отказе системы.

Fail Safe - это инструмент восстановления после катастрофического сбоя, поэтому на время выполнения операции перехвата управления данные оказываются недоступными. Начиная с версии Oracle9i для повышения доступности сервера рекомендуется применять подсистему Real Application Clusters.

### **Oracle Real Application Clusters**

В версии Oracle9i на смену подсистеме Oracle Parallel Server (OPS) пришла технология Real Application Clusters (RAC). RAC поддерживает перехват управления при отказе, а также повышает степень масштабируемости на кластерных конфигурациях в системах UNIX, Linux и Windows. Ключом к повышению масштабируемости стал механизм Cache Fusion, который существенно уменьшает количество операций записи на диск. Ранее именно так работало управление блокировками данных. В версии Oracle Database 10g переносимость RAC была поднята на новый уровень за счет интегрированного «кластерного ПО» (clusterware) для всех поддерживаемых RAC платформ.

Подсистема Real Application Clusters позволяет развертывать несколько экземпляров Oracle на нескольких узлах кластера или решетки (grid). RAC координирует трафик между системами или узлами, так что все экземпляры функционируют как единая база данных. В результате база данных способна масштабироваться на десятки узлов. Поскольку кластер предоставляет нескольким экземплярам возможность доступа к одним и тем же данным, отказ одного экземпляра не вызовет заметных задержек на время восстановления системы. Достаточно просто перенаправить пользователей на другой, работающий экземпляр. Осуществить прозрачный для пользователя перехват управления приложения могут с помощью интерфейса уровня вызовов Oracle Call Interface (OCI).

### **Data Guard и RAC**

Начиная с версии Oracle9i сочетание Data Guard и RAC заменило технологию Parallel Fail Safe. При наличии RAC механизм Data Guard обеспечивает автоматический перехват управления с ограниченным временем восстановления. Кроме того, он перенаправляет клиентов с отказавшего экземпляра на работающий, гарантируя быстрое установление нового соединения, и автоматически диагностирует состояние экземпляров.

### **Automated Storage Management**

В версии Oracle Database 10g появилась подсистема Automated Storage Management (ASM), которая обеспечивает оптимальное расслоение и зеркалирование данных для достижения максимальной производительности и доступности. Поскольку ASM управляется из программы Enterprise Manager, теперь администратор базы данных может сам выполнять это критически важное задание, не согласовывая свои действия с системным администратором.

### **Real Application Testing Option**

В версии Oracle Database 11g появилась возможность повторять все операции, выполненные в промышленной базе, и тестировать влияние изменений в системе. Это обеспечивает подсистема Real Application Testing Option. В нее входят средство Database Replay и программа SQL Performance Analyzer. Database Replay собирает информацию о рабочей нагрузке в промышленной базе, в том числе о конкуренции, зависимостях и временных затратах. Затем файлы данных о рабочей нагрузке преобразуются в файлы воспроизведения и передаются программе Replay Client для обработки. Кроме того, Database Replay предоставляет средства формирования отчетов о производительности и об имевших место ошибках. SQL Performance Analyzer получает подлежащие анализу данные о рабочей нагрузке, измеряет производительность до и после изменений в базе данных и показывает, как изменилось время выполнения каждой SQL-команды.

### **Средства обеспечения безопасности базы данных**

В Oracle имеются базовые средства безопасности для управления правами доступа

пользователей путем задания ролей и привилегий. Программа Enterprise Manager позволяет выполнять это локально или глобально. В последнем случае используется механизм безопасности уровня предприятия, являющийся частью подсистемы Advanced Security Option. Имеющиеся в Oracle средства обеспечения безопасности позволяют реализовать виртуальную частную базу данных (Virtual Private Database, VPD) путем создания политик и присоединения их к таблицам, представлениям или синонимам. Выполнение политик обеспечивается путем добавления предикатов к предложению WHERE команды SELECT, INSERT, UPDATE, DELETE или INDEX.

Во многих организациях требуется более строгая защита данных, хотя в наши дни доступ к базе данных может производиться из точек за пределами организации. Корпорация Oracle включила в СУБД средства безопасного развертывания и в таких сложных условиях. Речь идет о подсистемах Advanced Security Option, Label Security Option, Database Vault и Audit Vault.

## **Advanced Security Option**

Подсистема Advanced Security Option раньше называлась Advanced Networking Option (ANO). Основные механизмы обеспечения повышенной безопасности Oracle Net - это шифры RC4 (разработка компании RSA Data Security), Data Encryption Standard (DES), Triple DES и Advanced Encryption Standard (AES).

Для аутентификации можно применять систему Kerberos, RADIUS или Distributed Computing Environment (DCE). Для проверки целостности данных применяются алгоритмы MD5 и SHA-1. В версии Oracle Database 11g добавилось прозрачное шифрование данных и расширенная аутентификация посредством Kerberos с применением типов шифрования Oracle.

## **Label Security Option**

Подсистема Label Security Option управляет доступом к данным, сравнивая метки, сопоставленные строкам данных, с правами доступа к меткам, которые хранятся в привилегиях пользователя. В одной базе данных может быть несколько уровней авторизации. Авторизация меток задается в менеджере политик Policy Manager. Политики применяются к базовым объектам базы данных, а не к представлениям, что заметно упрощает задачу управления доступом к данным и повышает степень безопасности.

## **Database Vault Option**

Подсистема Database Vault Option обеспечивает детальное управление доступом к данным со стороны любого пользователя, включая и администраторов базы данных. Администратор по безопасности может задать условия, определяющие возможность доступа к базе, и проводить аудит различных аспектов безопасности. На более детальном уровне можно определить области (realm), чтобы разрешить доступ только конкретным ролям или лишь к определенным схемам.

## **Audit Vault Server**

Сервер аудита Oracle Audit Vault Server ведет мониторинг таблиц аудита в базе данных, журналов и управляющих файлов операционной системы, отслеживая подозрительные действия. Он может генерировать отчеты и отправлять оповещения при регистрации необычной активности.

## **Инструменты разработки Oracle**

В распоряжении разработчиков имеется много инструментов, позволяющих представлять данные и создавать более сложные приложения для работы с базой данных Oracle. Хотя эта книга посвящена собственно СУБД Oracle, мы кратко опишем основные инструменты, которые Oracle предлагает для разработки приложений: Oracle JDeveloper, Oracle SQL Developer и Oracle Developer Suite. Комплект Developer Suite, который иногда называют Oracle Internet Developer Suite, включает программы Oracle Forms Developer, Oracle Reports Developer, Oracle Designer, Oracle Discoverer Administrative Edition и Oracle Portal.

## **Oracle JDeveloper**

Oracle представила программу Oracle JDeveloper в 1998 году. Она позволяет разрабатывать простые приложения на языке Java без написания кода. Сейчас JDeveloper распространяется бесплатно, ее можно загрузить с сайта Oracle Technology Network. В нее входят: мастер форм данных Data Form Wizard, мастер Beans Express Wizard для создания компонентов JavaBeans и классов BeanInfo и мастер развертывания Deployment Wizard. JDeveloper включает также средства для работы с базой данных: различные драйверы для Oracle, редактор соединений Connection Editor, позволяющий скрыть сложность JDBC API, компоненты для привязки визуальных элементов управления к данным и прекомпилятор SQLJ, позволяющий встраивать

в код на Java команды SQL для доступа к базе данных. Приложения, разработанные на JDeveloper, можно развертывать на сервере приложений Oracle Application Server. Хотя мастера JDeveloper позволяют программисту создавать Java-объекты без какого-либо кодирования, конечным результатом все же является сгенерированный код на Java.

### **Oracle SQL Developer**

Программа Oracle SQL Developer была представлена в 2006 году. Она позволяет соединяться с любой базой данных Oracle версии не ниже Oracle9i Release 2. SQL Developer умеет создавать соединение с базой данных Oracle, показывать хранящиеся в базе объекты, создавать и модифицировать объекты в базе, запрашивать и обновлять данные, экспортировать данные и их описания, импортировать данные, обрабатывать команды, создавать и запускать отчеты. Входящие в состав продукта инструменты поддерживают редактирование, отладку и запуск PL/SQL-сценариев. Кроме того, SQL Developer может показывать объекты в базах данных других производителей и предоставляет средства для миграции на СУБД Oracle.

SQL Developer распространяется бесплатно, его можно загрузить с сайта Oracle Technology Network. Имеются версии для Windows, Linux и Apple Mac OS X. Кроме того, Oracle поддерживает на сайте Oracle Technology Network форум, посвященный SQL Developer.

### **Oracle Forms Developer**

Oracle Forms Developer - это инструмент создания диаграмм и приложений на базе форм, которые могут быть развернуты как традиционные клиент-серверные приложения или для работы в трехуровневой архитектуре. В последнем случае приложение выполняется в браузере и обращается к серверу приложений Oracle Application Server. Developer - это язык четвертого поколения (4GL). Приложение на таком языке пишется не в виде процедурного кода, а путем задания значений свойств. Developer поддерживает широкий спектр клиентов, в том числе написанных на Java. Программа Forms Builder включает встроенную виртуальную Java-машину для тестирования веб-приложений.

### **Oracle Reports Developer**

Программа Oracle Reports Developer предоставляет среду разработки и развертывания для быстрого построения и публикации отчетов в Сети с помощью системы Reports for Oracle Application Server. Данные могут быть представлены в виде таблиц, матриц, отчетов с группировкой, графиков или сочетания всего перечисленного. Высокое качество презентации достигается с помощью каскадных таблиц стилей (CSS).

### **Oracle Designer**

Программа Oracle Designer представляет собой графическую систему быстрой разработки приложений (Rapid Application Development, RAD), охватывающую весь процесс создания приложения для работы с базой данных - от построения бизнес-модели до проектирования схемы, генерации и развертывания. Проекты и изменения хранятся в многопользовательском репозитории. Инструмент позволяет выполнять реинжиниринг имеющихся таблиц и схем из баз данных как Oracle, так и других производителей, для повторного использования и перепроектирования.

Designer включает также генераторы приложений для Oracle Developer, HTML-клиентов, обращающихся к Oracle Application Server, и на языке C++. Designer может генерировать новые приложения и реконструировать имеющиеся приложения, в том числе модифицированные. Это позволяет реализовать процесс *кругового конструирования* (round-trip engineering), когда разработчик сначала генерирует приложение с помощью Designer, потом модифицирует его, реконструирует и помещает изменения обратно в репозиторий Designer.

### **Oracle Discoverer Administration Edition**

Программа Oracle Discoverer Administration Edition позволяет настроить и администрировать уровень Discoverer End User Layer (EUL), принадлежащий предыдущему поколению инструментов бизнес-анализа для Oracle. Назначение этого уровня - оградить от сложности SQL бизнес-аналитиков, использующих Discoverer как инструмент для выполнения произвольных запросов и анализа результатов. На всем протяжении процедуры построения EUL администратору помогают мастера. Кроме того, администратор может ограничить ресурсы, доступные аналитикам; за превышением квот будет следить входящий в Discoverer менеджер запросов.

### **Oracle Portal**

Oracle Portal был выпущен в 1999 под названием WebDB. Это основанный на HTML инструмент

разработки веб-приложений и сайтов, управляемых контентом. Портальные приложения развертываются в браузере. В состав Portal входят мастера для разработки компонентов приложения, инкапсулирующих сервлеты, для доступа к другим сайтам по протоколу HTTP. Разрабатываемые порталы допускают настройку под конкретного пользователя и развертываются на промежуточном слое в составе Oracle Application Server.

Oracle Portal привнес в WebDB важное усовершенствование - возможность создания и использования *портлетов*, позволяющих разбить веб-страницу на отдельные области, способные отображать информацию и взаимодействовать с пользователем независимо друг от друга. Например, из портлетов можно независимо обращаться к компонентам Answers, Discoverer и Reports.

Следующий продукт Oracle, реализующий инфраструктуру для создания порталов, - WebCenter - был выпущен в 2006 году и первоначально поставлялся как дополнительный компонент к Application Server.

## **Встраиваемые базы данных**

Семейство СУБД Oracle можно использовать во встраиваемых приложениях, но потребление памяти может оказаться недопустимо большим, а функциональность частично излишней. Сегодня Oracle предлагает другие встраиваемые базы данных, в том числе TimesTen, Berkeley DB и Oracle Database Lite. Они специально написаны так, что потребляют относительно мало ресурсов, и предназначены для других целей. Поэтому мы лишь кратко опишем их ниже и больше возвращаться к ним в этой книге не будем.

### **Oracle TimesTen**

Oracle TimesTen - это реляционная база данных, которая находится целиком в физической памяти и обычно применяется для высокопроизводительной обработки транзакций. Доступ к данным, хранящимся в TimesTen, осуществляется посредством SQL, JDBC, JMS и ODBC. База данных под управлением TimesTen может работать в режиме монопольного или разделяемого доступа и создаваться как постоянная или временная.

Обновление базы данных производится путем сбора данных с помощью библиотек TimesTen, скомпонованных с приложением, или из базы данных Oracle посредством механизма Cache Connect. Поскольку данные извлекаются из оперативной памяти и там же обновляются, среднее время считывания или обновления обычно составляет миллионные доли секунды. Механизм Cache Connect поддерживает кэширование данных, полученных из базы Oracle, как при чтении, так и при записи. Синхронизация TimesTen и Oracle может быть двусторонней.

Как и положено встраиваемым базам данных, TimesTen почти не требует администрирования. Возможна репликация из одной базы данных TimesTen в другую с помощью дополнительных средств, причем по умолчанию это делается асинхронно.

### **Oracle Berkeley DB**

Oracle Berkeley DB - это встраиваемый движок базы данных, потребляющий очень мало ресурсов и обеспечивающий блокировку на уровне записей. Поставляется в виде версий для Java и XML. База спроектирована для работы в одном процессе с приложением. Если Berkeley DB развертывается в таком режиме, то никакого отдельного администрирования базы данных вообще не требуется. Для работы может хватить всего 400 Кбайт.

В редакции Berkeley DB Java Edition поддерживаются Java Transaction API (JTA), J2EE Connector Architecture (JCA) и Java Management Extensions (JMX). Продукт в этом случае представляет собой единственный JAR-файл размером 820 Кбайт и работает в контексте той же виртуальной Java-машины, что и само приложение. Для доступа к Java-объектам предназначен слой Direct Persistence Layer (DPL).

Редакция Berkeley DB XML Edition чаще всего применяется в сетевых приложениях для управления контентом. Поддерживаются языки XQuery и Xpath.

Обе редакции можно сконфигурировать для обеспечения высокой доступности за счет репликации. Также поддерживается автоматическое восстановление. Решение о таком способе развертывания принимается на этапе проектирования приложения.

### **Oracle Lite**

Oracle Lite - это семейство продуктов для разработки мобильных приложений, нуждающихся в базе данных. Основные компоненты - Oracle Lite Database, Mobile Development Kit и Mobile Server (расширение Oracle Application Server).

Для ядра Oracle Lite Database требуется от 50 Кбайт до 1 Мбайт памяти в зависимости от платформы. Обращаться к базе можно с помощью языков Mobile SQL, C++ и Java. Также



поддерживается интерфейс ODBC. Поддержка Java включает написанные на Java хранимые процедуры и интерфейс JDBC. Оптимизация и администрирование Oracle Lite Database производятся автоматически. Эта СУБД может работать на карманных устройствах под управлением операционных систем Windows CE, Symbian, Windows и Linux.

Обычно при работе с Oracle Lite пользователь подключает свое карманное или мобильное устройство, в котором установлена база данных Oracle Lite Database, к полноценному серверу Oracle Database Server. После этого происходит автоматическая синхронизация данных между двумя системами. Затем пользователь может отключить устройство от сети и работать в автономном режиме. Сделав все необходимое, он снова подключается к серверу и синхронизирует данные.

Oracle Lite поддерживает различные механизмы синхронизации:

- двусторонняя синхронизация между мобильным устройством и сервером Oracle;
- синхронизация на базе модели «издатель-подписчик»;
- поддержка протоколов TCP/IP, HTTP, CDPD, 802.1 и HotSync.

Можно задать репликацию подмножеств данных с разными приоритетами. Поскольку нахождение данных в разных точках может приводить к конфликтам (в каком месте находится «правильная» версия?), предоставляется механизм автоматического разрешения конфликтов, допускающий и ручную настройку.

Mobile Server предоставляет единую платформу для публикации, развертывания, синхронизации и управления мобильными приложениями. Для контроля доступа к мобильным приложениям можно использовать развернутый в Сети центр управления. Кроме того, в состав Mobile Server вошел прежний продукт Oracle «Web-to-Go», который обеспечивает централизованный, управляемый мастерами механизм разработки и развертывания приложений.

## Лекция 2. Архитектура Oracle.

### Базы данных и экземпляры

Многие пользователи Oracle употребляют термины *экземпляр* и *база данных* как синонимы. На самом деле это разные (хотя и взаимосвязанные) вещи. Различие существенно, так как проливает свет на архитектуру Oracle.

В Oracle термином *база данных* описывается физическое хранилище информации, а термином *экземпляр* - программное обеспечение, работающее на сервере и предоставляющее доступ к информации в базе данных. Экземпляр исполняется на конкретном компьютере или сервере; база данных хранится на дисках, подключенных к этому серверу.

База данных - *физическая* сущность: она состоит из файлов, хранящихся на дисках. Экземпляр-сущность *логическая*: он состоит из структур в оперативной памяти и процессов, работающих на сервере. Например, Oracle использует область разделяемой памяти System Global Area (SGA, системная глобальная область) и области памяти в каждом процессе - Program Global Area (PGA, программная глобальная область). Экземпляр может быть частью одной и только одной базы данных. Напротив, с одной базой данных может быть ассоциировано несколько экземпляров. Время жизни экземпляров ограничено, тогда как база данных при должном обслуживании может существовать вечно.

Пользователи не имеют прямого доступа к информации, хранящейся в базе данных Oracle; они должны запрашивать информацию у экземпляра Oracle.

В реальном мире есть хорошая аналогия экземплярам и базам данных. Можно считать экземпляр мостом к базе данных, а саму ее - островом. Транспорт попадает на остров и уходит с него по мосту. Если мост перекрыт, то остров на месте, но транспорту туда не попасть. В терминологии Oracle, если экземпляр запущен, то данные могут попадать в базу и уходить из нее. Физическое состояние базы данных при этом изменяется. Если же экземпляр остановлен, то пользователи не могут обращаться к базе данных, пусть даже физически она никуда не делась. База данных в этом случае статична, никаких изменений в ней не происходит. Экземпляр снова запущен - и данные тут как тут.

### Структура базы данных Oracle

База данных состоит из табличных пространств, управляющих файлов, журналов, архивных журналов, файлов трассировки изменения блоков, ретроспективных журналов и файлов резервных копий (RMAN).

#### Табличные пространства

Любые данные, хранящиеся в базе Oracle, должны находиться в каком-то табличном пространстве. *Табличное пространство* (tablespace) - это логическая структура; нельзя попросить операционную систему показать вам табличное пространство. Каждое табличное

пространство состоит из физических структур, называемых *файлами данных* (data- files). В одном табличном пространстве может быть один или несколько файлов данных, тогда как каждый файл данных принадлежит ровно одному табличному пространству. При создании таблицы можно указать, в какое табличное пространство ее поместить. Тогда Oracle найдет для нее место в одном из файлов данных, составляющих указанное табличное пространство.

На рис. 2.2 показано соотношение между табличными пространствами и файлами данных.

Здесь мы видим два табличных пространства в базе данных Oracle. При создании новой таблицы ее можно поместить в табличное пространство DATA1 или DATA2. Физически таблица окажется в одном из файлов данных, составляющих указанное табличное пространство.

Начиная с версии Oracle Database 10g Release 2 для всех типов таблиц по умолчанию подразумеваются *локально управляемые табличные пространства*. В таком табличном пространстве можно создавать *большие файлы*, то есть при работе в 64-разрядных системах задействуется возможность создавать сверхбольшие файлы.

В Oracle9i появился механизм файлов, управляемых Oracle (Oracle Managed Files, OMF), позволяющий автоматически создавать, именовать и, если понадобится, удалять все файлы, составляющие базу данных. OMF упрощает обслуживание базы данных, поскольку не нужно помнить имена всех составляющих ее файлов. К тому же не возникают проблемы из-за ошибок человека, ответственного за именование файлов. Начиная с версии Oracle Database 10g сочетание OMF и табличных пространств с большими файлами делает работу с файлами данных совершенно прозрачной.

Максимальное количество файлов данных в базе Oracle- 64 000. Поскольку табличное пространство с большими файлами может содержать файл, который в 1024 раза больше файла в табличном пространстве с малыми файлами, а размер блока в табличном пространстве с большими файлами для 64-разрядных операционных систем составляет 32 Кбайт, общий размер базы данных Oracle может достигать 8 экзабайт (1 экзабайт = 1 000 000 терабайт). Табличные пространства с большими файлами предназначены для использования совместно с подсистемой автоматического управления хранением Automatic Storage Management (ASM), иными менеджерами логических томов, поддерживающими расслоение, и RAID-массивами.

## Файлы базы данных

База данных Oracle состоит из физических файлов трех основных типов:

- управляющие файлы (control files);
- файлы данных (datafiles);
- журнальные файлы, или журналы (redo log files).

В управляющем файле хранится информация о местонахождении других физических файлов, составляющих базу данных, - файлов данных и журналов. Там же хранится важная информация о содержимом и состоянии базы данных:

- имя базы данных;
- время создания базы данных;
- имена и местонахождение файлов данных и журнальных файлов;
- информация о табличных пространствах;
- информация о файлах данных в автономном режиме;
- история журналов и информация о порядковом номере текущего журнала;
- информация об архивных журналах;
- информация о наборах и фрагментах резервных копий, файлах данных и журналах;
- информация о копиях файлов данных;
- информация о контрольных точках.

Управляющие файлы не только содержат важную информацию, необходимую при запуске экземпляра, они полезны и при удалении базы данных. Начиная с версии Oracle Database 10g с помощью команды DROP DATABASE можно удалить все файлы, перечисленные в управляющем файле базы данных, а также сам управляющий файл.

## Инициализация базы данных

При запуске экземпляра Oracle считываются параметры инициализации. Они определяют, как база данных должна использовать физическую инфраструктуру и иную конфигурационную информацию об экземпляре. Параметры инициализации хранятся в файле параметров инициализации экземпляра, который обычно называют просто *INIT.ORA* или, начиная с версии Oracle9\*, в репозитории, который называется файлом параметров сервера (или *SPFILE*).

Количество обязательных параметров инициализации уменьшается с выходом каждой новой версии Oracle. В дистрибутиве Oracle есть пример файла инициализации, пригодный для запуска базы данных. Либо можно воспользоваться программой Database Configuration Assistant (DCA), которая подскажет обязательные значения (например, имя базы данных).

## Развертывание физических компонентов

Этот раздел не может заменить официальную документацию по установке Oracle. Наша цель - снабдить вас практическим руководством по планированию развертывания базы данных Oracle.

## Управляющие файлы

Для любой базы данных следует хранить по меньшей мере два управляющих файла на разных физических дисках. Без актуальной копии управляющего файла вы рискуете потерять информацию о составляющих вашей базы данных. Утрата управляющих файлов не обязательно фатальна, их можно и воссоздать. Но процедура воссоздания довольно сложна и рискованна, а избежать ее несложно.

Местоположение управляющих файлов определяется, как уже было сказано, параметром инициализации `CONTROL_FILES`. Он позволяет задать несколько управляющих файлов, например:

```
control_files = (/u00/oradata/control.001.dbf,  
                /u01/oradata/control.002.dbf,  
                /u02/oradata/control.003.dbf)
```

Этот параметр сообщает экземпляру, где искать управляющие файлы. Oracle гарантирует, что все копии управляющего файла одинаковы, то есть любые изменения вносятся синхронно. Если параметр не задан, Oracle создаст управляющий файл с именем по умолчанию или прибегнет к услугам компонента Oracle Managed Files (если тот активирован).

Многие базы данных Oracle развертываются на том или ином варианте RAID-массива, например RAID-1 или RAID-5, чтобы избежать потери данных в случае выхода диска из строя. Напрашивается вывод, что можно обойтись без нескольких копий, сохранив управляющий файл в защищенной дисковой памяти, и что утрата диска еще не означает утраты управляющего файла. Но этот вывод неправомерен по двум причинам:

1. Если в массиве с расслоением (*striped array*) или в зеркальной паре (*mirror-pair*) отказывает больше одного диска, то все данные, хранящиеся на этих дисках, теряются. Статистически это редкое событие, но все же такое случается, и тогда есть угроза повреждения или утраты управляющего файла. Поскольку вы и так будете по горло заняты восстановлением после множественных сбоев диска, вероятно, лучше при этом избежать хотя бы воссоздания управляющих файлов. Создание дополнительных копий, пусть даже хранящихся в избыточной дисковой памяти, - это дополнительный уровень защиты.

2. Избыточная дисковая память не поможет защититься от человеческих ошибок. Кто-то может случайно удалить или переименовать управляющий файл, затереть его другим или переместить в другое место. И зеркалированный диск честно отразит эти изменения. А при резервировании управляющих файлов хотя бы одна копия да останется.

Не стоит беспокоиться о том, что запись в несколько управляющих файлов понизит производительность. Обновление управляющих файлов - ничто по сравнению с другими операциями дискового ввода/вывода, производимыми Oracle.

## Файлы данных

В файлах данных находятся собственно данные, хранящиеся в базе: таблицы и индексы, словарь данных, в котором сохраняется информация об этих структурах, и сегменты отката, необходимые для реализации конкурентного доступа.

## Структура файла данных

Первый блок файла данных называется *заголовком файла данных*. В нем хранится важная информация, необходимая для поддержания целостности базы данных, в частности *структура контрольной точки*. Она представляет собой логическую временную метку, показывающую, когда в последний раз были записаны изменения в файл данных. Эта информация абсолютно необходима для процесса восстановления, поскольку определяет, какие журналы нужно накатить, чтобы привести файл данных к состоянию на текущий момент времени.

## Журнальные файлы

Журнальный файл содержит протокол всех изменений, произведенных в базе данных в результате выполнения транзакций и внутренних операций Oracle. Обычно измененные блоки кэшируются в памяти, поэтому в случае сбоя экземпляра может оказаться, что какие-то блоки не записались в файлы

данных. Тогда можно воспользоваться протоколом, хранящимся в журнальных файлах, чтобы воспроизвести изменения, оставшиеся не записанными в момент сбоя, и тем самым обеспечить согласованность транзакции.

Кроме того, журнальные файлы применяются для реализации операций отката (undo), инициируемых командой ROLLBACK. Незафиксированные изменения в базе данных откатываются, так что база остается в том состоянии, в котором находилась в момент последней фиксации.

Чтобы упростить работу в случае сбоя, рекомендуем после любой не зарегистрированной в журнале операции снимать резервную копию, если вы не можете позволить себе потерять созданный объект или по какой-то причине операцию невозможно повторить. Помимо использования слова NOLOGGING в отдельных командах, можно пометить таблицу или все табличное пространство атрибутом NOLOGGING. Тогда запись в журнал будет подавляться для всех операций с данной таблицей или с любой таблицей в помеченном табличном пространстве.

### **Как Oracle использует журнальные файлы**

Заполнив один журнальный файл, Oracle автоматически переходит к следующему. Заполнив все имеющиеся журнальные файлы, Oracle

возвращается к первому и перезаписывает его. Для отслеживания журналов применяется порядковая нумерация. Порядковый номер записывается в текущий журнальный файл.

Чтобы лучше разобраться в именах журнальных файлов и их порядковых номерах, рассмотрим три таких файла с именами *redolog1.log*, *redolog2.log*, *redolog3.log*. Когда Oracle только начинает их использовать, журналам присвоены порядковые номера 1, 2 и 3 соответственно. После того как Oracle вернется к первому журналу - *redolog1.log*- и начнет перезаписывать его, журналу будет присвоен порядковый номер 4. Затем настанет очередь файла *redolog2.log*, который получит порядковый номер 5.

Помните, что операционная система идентифицирует журнальные файлы по имени, а Oracle определяет, в каком порядке заполнялись журналы, ориентируясь на порядковый номер. Поскольку журналы автоматически используются повторно, имя файла мало что говорит о его месте в последовательности.

На рис. 2.1 проиллюстрировано заполнение журналов и их циклический перебор.

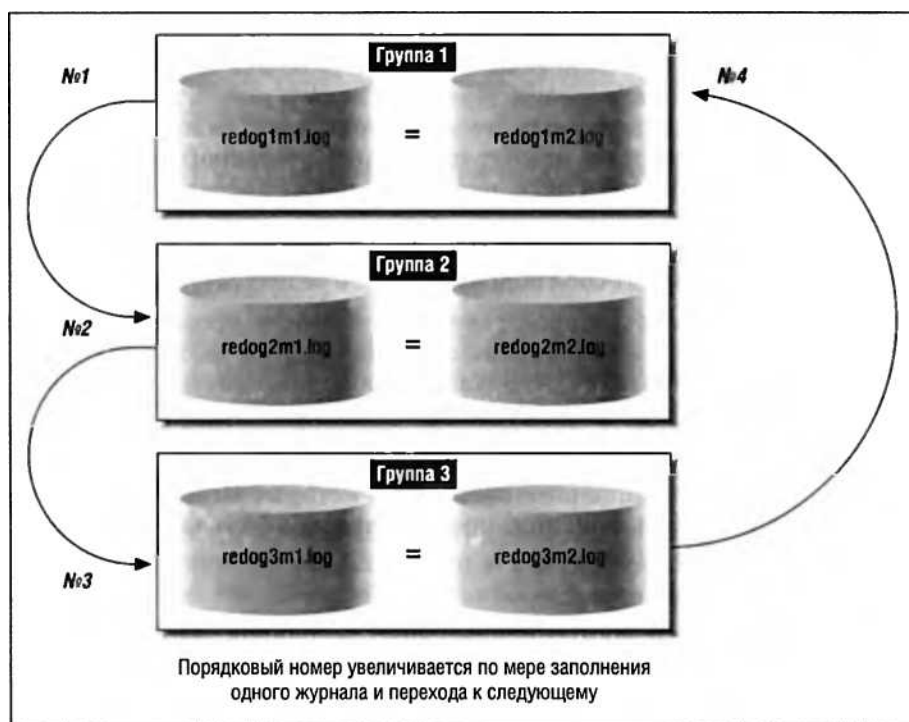


Рис. 2.1. Циклический перебор журналов

### Соглашения об именовании журнальных файлов

Имена различных файлов, составляющих базу данных, очень важны - по крайней мере, для тех, кто порой вынужден искать файлы по именам. Если вы не пользуетесь компонентом Oracle Managed Files, то должны выработать схему именования, отражающую назначение и существенные характеристики файла. Одно из возможных соглашений о выборе имен журнальных файлов показано на рис. 2.1:

*redog1m1.log*, *redog1m2.log*, ...

Префикс *redo* и суффикс *.log* отражают тот факт, что файл содержит журнал. В строках *g1m1* и *g1m2* закодирована информация о номерах группы и элемента. Это всего лишь пример, но мы рекомендуем принять какое-то соглашение, которое кажется вам осмысленным, и строго придерживаться его.

### Архивные журнальные файлы

Возможно, вас интересует, как избежать потери критически важной информации, коль скоро Oracle возвращается к ранее записанному журналу.

Есть два подхода к этой проблеме. Первый чрезвычайно прост: вы даже не пытаетесь избежать утраты информации в случае сбоя - со всеми вытекающими последствиями. При перезаписи журнала хранящаяся в нем история изменений теряется. Если в результате сбоя будут повреждены файлы данных, то можно будет восстановить всю базу данных в состоянии на момент последнего резервного копирования. Но воспроизвести изменения, внесенные позже последнего резервного копирования, без журнала не удастся. Таким путем идут очень немногие компании, работающие с Oracle, ведь невозможность восстановить ситуацию на момент сбоя недопустима - в результате теряются данные.

Второе (и более распространенное) решение проблемы заключается в архивировании заполненных журналов. Для того чтобы понять, что такое архивирование журналов, необходимо знать, что Oracle на самом деле поддерживает два типа журналов:

### *Оперативные журналы*

Файлы операционной системы, в которые Oracle последовательно записывает изменения, произведенные в базе данных, циклически перебирая файлы.

### *Архивные журналы*

Копии заполненных оперативных журналов, создаваемые во избежание потери данных при перезаписи оперативных журналов.

СУБД Oracle может работать в одном из двух режимов архивирования журналов:

#### **NOARCHIVELOG**

Как видно из названия, в этом режиме журналы не архивируются. По мере циклического перебора заполненные журналы повторно инициализируются и перезаписываются, при этом история изменений базы данных стирается. Именно о таком режиме шла речь выше: в этом случае сбой приводит к невозможной потере данных.

Тем, кто выбирает работу без архивирования журналов, будет доступно значительно меньшее количество вариантов и параметров резервного копирования базы данных (см. главу 11). Корпорация Oracle не рекомендует этот режим.

#### **ARCHIVELOG**

Переходя к новому журналу, Oracle архивирует предыдущий. Для того чтобы не допустить появления пропусков в истории изменений, повторное заполнение журнала начинается только после его успешного архивирования. Архивные и оперативные журналы содержат полную историю изменений, произведенных в базе данных. В совокупности они позволяют серверу восстановить все зафиксированные транзакции вплоть до момента сбоя. При работе в этом режиме возможно резервное копирование табличных пространств и отдельных файлов данных.

Использовать оперативные и архивные журналы на этапе восстановления базы данных серверу помогают вышеупомянутые внутренние порядковые номера.

### **Режим ARCHIVELOG и автоматическое архивирование**

Начиная с версии Oracle Database 10g\* можно перевести БД в режим ARCHIVELOG командой:

```
ALTER DATABASE ARCHIVELOG
```

Если база данных работает в режиме ARCHIVELOG, то, заполнив очередной журнал, Oracle помечает его как подлежащий архивированию. Прежде чем заполненный журнал можно будет использовать повторно, его необходимо архивировать. Команда ALTER DATABASE ARCHIVELOG по умолчанию включает режим автоматического архивирования и запускает процессы-архиваторы.

До выхода версии Oracle Database 10g\* пометка файла журнала как готового к архивированию еще не означала, что он будет архивирован автоматически. Необходимо было также задать параметр в файле инициализации:

```
LOG_ARCHIVE_START = TRUE
```

Только в этом случае запускался процесс копирования заполненного оперативного журнала в архивный.

Местоположение и формат имен архивных журналов определяются еще двумя параметрами: LOG\_ARCHIVE\_DEST и LOG\_ARCHIVE\_FORMAT. Например, строка

```
LOG_ARCHIVE_DEST = C:\ORANT\DATABASE\ARCHIVE
```

определяет, в какой каталог Oracle будет записывать архивные журналы, а строка

```
LOG_ARCHIVE_FORMAT = ORCL%t_%s_%r.arc
```

описывает формат имен файлов архивных журналов. В данном случае имена начинаются с префикса ORCL и заканчиваются суффиксом .arc. В форматной строке распознаются следующие спецификаторы:

**%t**

Включать в имя файла номер цепочки журнальных файлов.

**%s**

Включать в имя файла порядковый номер журнала.

**%r**

Включать в имя файла идентификатор сброса порядкового номера журнала.

Если вы хотите, чтобы номер цепочки, порядковый номер журнала и идентификатор сброса номеров в именах архивных журнальных файлов дополнялись нулями, то следует записывать спецификаторы заглавными буквами, например:

```
LOG_ARCHIVE_FORMAT = "ORCL%T_%S_%R.arc"
```

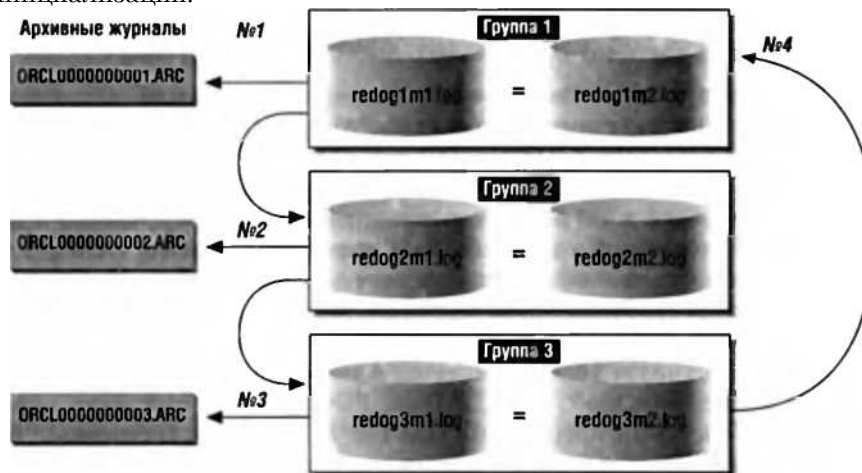
Поскольку файл инициализации считывается в момент запуска экземпляра Oracle, любые изменения вступают в силу только после останова и перезапуска экземпляра. Однако не забывайте, что включение режима автоматического архивирования еще не переводит базу данных в режим ARCHIVELOG. И

обратно, перевод базы данных в режим ARCHIVELOG не включает режим автоматического архивирования.

Следует также убедиться, что в файловой системе достаточно места для размещения архивных журналов. Если эта файловая система переполнится, то Oracle зависнет, не имея возможности архивировать журнальные файлы.

На рис. 2.2 показан порядок использования журналов, когда архивирование включено.

Архивные журналы исключительно важны для восстановления базы данных. Как и для оперативных, для архивных журналов можно задать несколько мест расположения. В этом случае Oracle будет копировать заполненные журналы в указанные места. Можно также указать, следует ли ожидать удачного завершения копирования всех журналов. Эта функциональность управляется следующими параметрами инициализации.



При заполнении одного журнала и переходе к следующему заполненный журнал архивируется  
**Рис. 2.2. Циклический перебор журналов с архивированием**

#### **LOG\_ARCHIVE\_DUPLEX\_DEST**

Задаёт дополнительные места для копий архивных файлов.

#### **LOG\_ARCHIVE\_MIN\_SUCCEED\_DEST**

Определяет, должно ли успешно завершиться копирование всех или хотя бы нескольких копий заполненного журнала. Допустимы значения от 1 до 10, если используется мультиплексирование, и 1 или 2 в случае дуплексного режима.

Дополнительные параметры и представления, управляющие описанной функциональностью, описаны в документации Oracle.

## **Память и процессы экземпляра**

Экземпляр Oracle можно определить как область разделяемой памяти и набор фоновых процессов. Область разделяемой памяти экземпляра называется *системной глобальной областью* (System Global Area, SGA). Фактически SGA является не одной большой однородной областью памяти, а состоит из различных компонентов, описанных в следующем разделе. Все процессы экземпляра, как системные, так и пользовательские, совместно обращаются к SGA.

До версии Oracle9i размер SGA устанавливался при запуске экземпляра Oracle. Единственным способом изменения размера SGA или какой-то ее составляющей было изменение соответствующих параметров инициализации, остановка и перезапуск экземпляра. В Oracle9i можно изменять размер SGA и ее компонентов, не останавливая экземпляр. В Oracle9i также введено понятие *гранулы*, то есть наименьшего объема памяти, который можно добавить или удалить из SGA.

В версии Oracle Database 10g появился механизм автоматического управления разделяемой памятью (Automatic Shared Memory Management, ASMM), а в Oracle Database 11g — механизм автоматического управления памятью (Automatic Memory Management, AMM) для компонентов SGA и PGA. Если задан параметр инициализации MEMORY\_TARGET (появился в Oracle Database 11g) или SGA\_TARGET, то база данных автоматически распределяет память между различными компонентами SGA, обеспечивая оптимальное управление памятью. К автоматически распределяемым компонентам относятся разделяемый пул (его размер вручную устанавливается с помощью параметра SHARED\_POOL\_SIZE), большой пул (LARGE\_POOL\_SIZE), пул Java (JAVA\_POOL\_SIZE), кэш буферов (DB\_CACHE\_SIZE) и пул Streams (STREAMS\_POOL\_SIZE). Параметры инициализации, относящиеся к автоматическому управлению памятью, можно задать в Oracle Enterprise Manager.

Фоновые процессы взаимодействуют с операционной системой и между собой, управляя структурами памяти экземпляра. Эти процессы также управляют собственно базой данных на диске и выполняют общие действия по обслуживанию экземпляра. На рис. 2.3 показаны структуры памяти и фоновые процессы, рассматриваемые в следующем разделе.

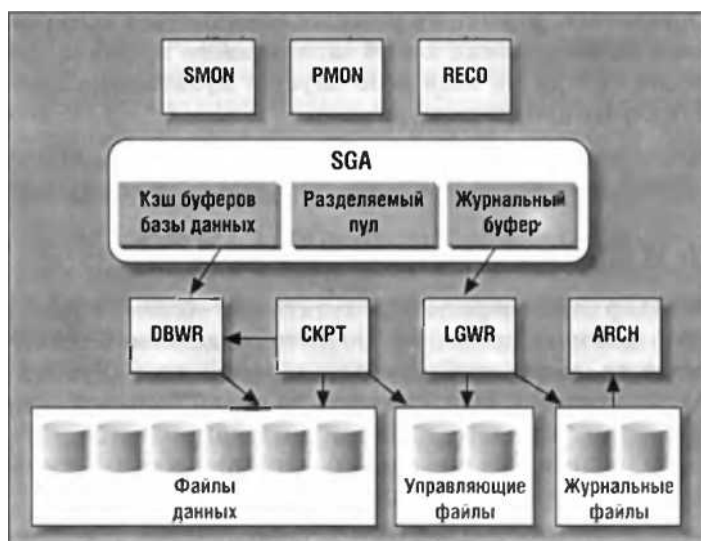


Рис. 2.3. Экземпляр Oracle

Если активированы еще какие-то компоненты СУБД, то могут работать и другие фоновые процессы, например: разделяемые серверы (до версии Oracle9i - Multi-Threaded Server, MTS), очереди заданий или репликация.

## Структуры оперативной памяти экземпляра

Как видно из рис. 2.3, системная глобальная область (SGA) состоит из нескольких областей. На рисунке показаны кэш буферов базы данных, разделяемый пул и журнальный буфер. Еще могут присутствовать пул Java, большой пул и пул Streams. Все эти области описаны ниже. Более подробно вопрос о связи SGA с производительностью рассмотрен в разделе «Как Oracle использует системную глобальную область» главы 7.

### Кэш буферов базы данных

В кэше буферов хранятся извлеченные из базы блоки данных. Такой буфер между пользовательскими запросами и реальными файлами данных повышает производительность СУБД Oracle. Если часть данных присутствует в кэше буферов (например, в результате недавно выполненного запроса), ее можно извлечь из оперативной памяти без затрат на обращение к диску. Oracle управляет кэшем на основе алгоритма LRU (Least Recently Used), когда выталкиваются элементы, не использовавшиеся дольше всех. Другими словами, если пользователь запрашивает данные, обращение к которым осуществлялось недавно, то с большой вероятностью они содержатся в кэше буферов и могут быть возвращены немедленно, без повторного считывания с диска.

Если же запрашивается блок, отсутствующий в кэше, то его необходимо считать и загрузить в кэш. Когда пользователь модифицирует данные в блоке, изменения сначала записываются в блок, находящийся в кэше, а спустя некоторое время сбрасываются в файл данных, которому принадлежит блок. Поэтому пользователю не нужно дожидаться завершения операции записи измененных блоков на диск.

Идея откладывания операций ввода/вывода до тех пор, пока это не станет абсолютно необходимо, красной нитью пронизывает всю СУБД Oracle. Диски - самый медленный компонент вычислительной системы, поэтому чем меньше объем ввода/вывода, тем быстрее работает система. Откладывая выполнение некритичных операций ввода/вывода, Oracle достигает более высокой производительности.

## Фоновые процессы экземпляра

Чаще всего используемые фоновые процессы показаны на рис. 2.3, хотя их состав меняется от версии к версии. Ниже описаны некоторые процессы.

*Процесс записи в базу данных (Database Writer, DBWR)*



Записывает блоки данных из кэша буферов SGA в файлы данных, расположенные на диске. Для одного экземпляра Oracle может существовать до 20 процессов DBW, обрабатывающих ввод/вывод в различные файлы данных, откуда и обозначение DBW<sub>n</sub>. Большинство экземпляров обходится одним процессом DBW. DBW записывает блоки из кэша на диск по двум основным причинам:

- Для установки контрольной точки (то есть для обновления блоков файлов данных, с тем чтобы они «догнали» журналы). Oracle записывает в журнал информацию, необходимую для повторного выполнения транзакции после ее фиксации, а позже сохраняет сами блоки. Периодически Oracle устанавливает контрольную точку для того, чтобы привести содержимое файла данных в соответствие с информацией, записанной в журнал для зафиксированных транзакций.
- Когда необходимо прочитать запрошенные пользователем блоки, а в кэше буферов не осталось свободного места. Тогда на диск сбрасываются блоки, которые дольше всего не использовались. Запись блоков именно в таком порядке позволяет минимизировать влияние отсутствия блоков в кэше на производительность.

#### *Процесс записи в журнал (Log Writer, LGWR)*

Записывает информацию из журнального буфера в SGA во все копии текущего журнального файла на диске. Пока транзакция обрабатывается, необходимая для ее повторного выполнения информация хранится в журнальном буфере в SGA. После фиксации транзакции Oracle отправляет эту информацию на постоянное хранение, вызывая процесс LGWR для ее записи на диск.

#### *Системный монитор (System Monitor, SMON)*

Отвечает за общую исправность и безопасность экземпляра. SMON восстанавливает экземпляр при запуске после сбоя. Он же координирует доступ и реализует восстановление экземпляра, когда к базе данных обращаются сразу несколько экземпляров, как бывает при работе с Real Application Clusters. SMON также приводит в порядок смежные области свободного пространства в файлах данных, объединяя их, и избавляется от пространства, используемого для сортировки строк, когда в нем уже нет необходимости.

#### *Монитор процессов (Process Monitor, PMON)*

Следит за пользовательскими процессами, обращающимися к базе данных. В случае аварийного прекращения пользовательского процесса PMON отвечает за очистку оставшихся занятыми ресурсов (таких как оперативная память) и за снятие всех блокировок, установленных сбойным процессом.

#### *Архиватор (Archiver, ARCH)*

Читает заполненные файлы журнала и копирует их в один или несколько архивных журналов. Поддерживается до 10 процессов архивации, которым присваиваются HMeHaARC0-ARC9. По мере необходимости LGWR запускает дополнительные архиваторы в зависимости от нагрузки. Максимально разрешенное количество процессов архивации задается параметром инициализации LOG\_ARCHIVE\_MAX\_PROCESSES. По умолчанию этот параметр равен 2, необходимость изменять это значение возникает редко.

#### *Контрольная точка (Checkpoint, CKPT)*

Обновляет заголовки файлов данных после установки контрольной точки.

#### *Процесс восстановления (Recover, RECO)*

Автоматически очищает неудавшиеся и отложенные распределенные транзакции.

#### *Диспетчер (Dispatcher)*

Эти необязательные фоновые процессы запускаются, если развернут разделяемый сервер.

#### *Служба глобального кэша (Global Cache Service, LMS)*

Управляет ресурсами, необходимыми Real Application Clusters, а также координирует использование ресурсов несколькими экземплярами.

#### *Очередь заданий (Job Queue)*

Служба выполнения команд и процедур PL/SQL по заданному расписанию.

#### *Монитор очередей (Queue Monitor, QMON)*

Осуществляет мониторинг очередей сообщений подсистемы Oracle Streams. Поддерживается до 10 таких мониторов.

#### *Процессы подсистемы автоматического управления хранением (Automatic Storage Management, ASM)*

Процесс RBAL координирует перебалансировку работ для групп дисков. ORB<sub>n</sub> выполняет собственно перебалансировку. Процесс ASMB обеспечивает коммуникацию между базой данных и экземпляром ASM.

## Словарь данных

Во всех версиях Oracle присутствует набор *метаданных*, описывающих различные структуры данных, например определения таблиц и ограничений целостности. Совокупность таблиц и представлений, в которых хранятся метаданные, называется *словарем данных* Oracle. Всем компонентам, которые рассматриваются в настоящей главе, соответствуют какие-то системные таблицы и представления в словаре данных, полностью описывающие характеристики компонента. К этим таблицам и представлениям можно обращаться с помощью стандартных команд SQL. В табл. 2.1 показано, где в словаре данных можно найти информацию о каждом компоненте.

Таблицы словаря данных всегда находятся в табличном пространстве SYSTEM. Имена динамических таблиц начинаются с префиксов V\$ или GV\$ (данные в динамической таблице постоянно обновляются, отражая текущее состояние базы данных). Имена статических таблиц могут начинаться с одного из префиксов DBA\_, ALL\_ или USER\_, обозначающего область видимости представленных в таблице объектов.

## Лекция 3. Установка и запуск Oracle

### Установка Oracle

Вплоть до версии Oracle8Z программа установки Oracle для UNIX поставлялась в текстовом и графическом вариантах. Графический инсталлятор использовал библиотеку Motif и работал под управлением системы X Windows. Для платформы Windows NT поставлялась только графическая версия. Начиная с версии Oracle8i инсталлятор переписан на языке Java.

Инсталлятор Oracle - одна из первых программ, демонстрирующих преимущества переносимости Java: он выглядит и работает одинаково во всех операционных системах. Установка Oracle стала довольно простым делом - пользователю надо лишь несколько раз щелкнуть мышью и ответить на вопросы о требуемых функциях и опциях.

Корпорация Oracle приложила немало усилий к тому, чтобы еще упростить установку в версии Oracle Database 10g. И эту версию, и Oracle Database 11g можно установить меньше чем за 20 минут.

В процессе установки инсталлятор также запускает помощник конфигурирования сети Net Configuration Assistant и помощник конфигурирования базы данных Database Configuration Assistant, поэтому по завершении процедуры вы получаете работающий экземпляр Oracle.

Если по какой-то причине установка не завершена, то в файле протокола вы найдете команды, завершившиеся с ошибкой. Это поможет вам понять, в чем проблема, и после ее устранения выполнить оставшиеся команды самостоятельно.

Хотя процедура установки теперь одинакова для всех платформ, все же для каждой платформы есть особенности. Каждая версия Oracle Database Server поставляется со своим комплектом документации. Внего входит руководство по установке, замечания к версии (сюда включена информация, добавленная уже после публикации руководства по установке) и книга «Приступая к работе». Перед началом установки следует прочитать эти документы, поскольку в каждом имеются весьма ценные сведения об особенностях процедуры установки.

Например, вам предстоит заранее определиться с тем, где разместить корневой каталог Oracle и файлы данных. Эти вопросы подробно рассматриваются в документации. Помимо печатной документации на приобретенном вами носителе с программным обеспечением имеется документация в электронном виде. В ней вы найдете дополнительную информацию о СУБД и смежных продуктах.

Обычно руководство по установке вкладывается в ту же коробку, где находится компакт-диск. В руководстве приведены требования к системе (объем оперативной и дисковой памяти), действия, которые необходимо завершить до установки, указания по выполнению установки и замечания о переходе со старых версий Oracle на новую. Помните, что полная установка ПО не исчерпывается копированием программ на диск; необходимо еще сконфигурировать и запустить основные службы.

В старых версиях еще до начала установки Oracle вы должны были определиться со структурой каталогов и соглашением об именовании файлов, составляющих базу данных. Ясные, последовательные и хорошо спланированные соглашения - необходимое условие минимизации человеческих ошибок при администрировании операционной системы и базы данных. Ныне выбор имен в значительной степени автоматизирован. К наиболее важным решениям в этой области можно отнести:

- имена дисков или точек монтирования;
- структуры каталогов для ПО Oracle и файлов базы данных;
- имена файлов, составляющих базу данных: управляющих файлов, файлов данных и