## Тема: *Объекты и субъекты моделей информационной безопасности баз данных на примере СУБД Oracle*

Неотъемлемой частью любого проекта по созданию или оценке системы информационной безопасности баз данных является разработка модели безопасности данных. Наиболее распространенной парадигмой моделей обеспечения безопасности данных является субъектно-объектная абстракция. Элементы системы делятся на два класса сущностей: активные сущности — субъекты и пассивные сущности — объекты. В автоматизированной системе, реализованной на основе СУБД промышленного уровня, прообразом всех этих сущностей являются классы однотипных объектов, создаваемых и описываемых средствами языка описания данных (DDL). Сохраняя стилевое единство с предыдущими книгами автора, будем вместо термина «классы однотипных объектов» использовать термин «объекты».

В качестве рабочего примера рассмотрим объекты СУБД Oracle. Хорошо известно, что СУБД Oracle поддерживает реляционную модель данных, поэтому естественно, что к числу основных объектов базы данных относятся таблица и представление.

Таблица (TABLE) является базовой структурой реляционной модели. Как известно, вся информация в реляционной базе данных хранится в таблицах. Полное имя таблицы в базе данных состоит из имени схемы и собственно имени таблицы. Таблицы состоят из множества поименованных столбцов или атрибутов. Множество допустимых значений атрибута называют доменом значений или просто доменом. Множество допустимых значений столбца также может быть уточнено с помощью статических ограничений целостности. Таблицы могут быть связаны между собой отношениями ссылочной целостности. Таблица может быть пустой или состоять из одной или более строк значений атрибутов. Строки значений атрибутов таблицы называются также кортежами. Для однозначной идентификации строки в таблице служит идентификатор (ROWID) —указатель, имеющий специальный формат. В Огас1е8 появились вложенные таблицы (NESTED TABLES), которые позволяют объявить таблицу как тип значения столбца родительской таблицы. Подробно они рассмотрены в разделе «Объектные расширения в Огас1е8». Для повышения скорости доступа к данным таблица может быть индексно организована (INDEX-ORGANIZED TABLE). Физическое пространство для хранения данных таблицы выделяется частями, называемыми экстентами. Размеры начального и дополнительных экстентов определяются при создании таблицы.

Представление (VIEW) — это поименованная, динамически поддерживаемая сервером выборка из одной или нескольких таблиц. По сути, представление — это производное множество строк, которое является результатом выполнения некоторого запроса к базовым таблицам. В словаре данных хранится только определение представления, и когда в операторе SQL встречается название представления, Oracle обращается к словарю за определением и подставляет его в исходный запрос. Запрос, определяющий выборку, ограничивает видимые пользователем данные. Представления позволяют упростить сложные запросы и сделать более понятными их логику. Используя представления, администратор безопасности ограничивает доступную пользователю часть базы данных только теми данными, которые реально необходимы для выполнения его работы. Представления также можно использовать для поддержки приложений при изменении структуры таблицы. Например, при добавлении нового столбца в таблицу создать представление, его не включающее.

Синоним (SYNONYM)— это альтернативное имя или псевдоним объекта Oracle, который позволяет пользователям базы данных иметь доступ к данному объекту. Синоним может быть частным и общим. Общий (PUBLIC) синоним позволяет всем пользователям базы данных обращаться к соответствующему объекту по альтернативному имени. Характерным применением общих синонимов является сокрытие информации о схеме, в которой расположен объект. Наличие синонима позволяет обращаться к объекту по имени, которое является абсолютным в масштабе базы данных. Реальная привязка объекта к некоторой схеме при этом скрыта от пользователя или приложения.

Для управления эффективностью доступа к данным Oracle поддерживает следующие объекты: индекс, табличная область, кластер и хэшированный кластер.

**Индекс (INDEX)** — это объект базы данных, предназначенный для повышения производительности выборки данных. Индекс создается для столбцов таблицы и обеспечивает более быстрый доступ к данным за счет хранения указателей (ROWID) на месторасположение строк. При обращении к индексированному столбцу сервер по предъявляемому значению находит в индексе указатели на эти строки, а потом непосредственно обращается к ним. Если все требуемые значения столбцов имеются в индексе, обращение к таблице не происходит вовсе. Имеется несколько типов индексов — В\*Тгее (двоичное дерево, каждый узел которого содержит указатель на следующий и предыдущий), масочный индекс, индекс с реверсированным ключом, кластерный индекс. Подробнее о них рассказывается в разделе «Методы повышения производительности».

**Кластер (CLUSTER)** — объект, задающий способ хранения данных нескольких таблиц, содержащих информацию, обычно обрабатываемую совместно, например значения столбцов таблиц, часто участвующих в эквисоединениях. Строки таких таблиц, имеющие одинаковое значение в кластеризованных столбцах, хранятся в базе данных специальным образом: на логическом уровне — в нормализованном виде, а на физическом — в ненормализованном. Кластеризация столбцов таблиц позволяет уменьшить время выполнения выборки.

**Хэшированный кластер (HASH CLUSTER)** — объект, задающий способ организации таблиц, базирующийся на результатах хэширования их первичных ключей. Для получения данных из такого кластера запрашиваемое значение ключа обрабатывается хэш-функцией, полученное значение определяет, в каком блоке кластера хранятся данные.

**Табличная область (TABLESPACE)** — именованная часть базы данных, используемая для распределения памяти для таблиц, индексов и других объектов. В табличную область входит один или несколько файлов. Это предоставляет возможность гибко настроить хранение данных в зависимости от порядка и интенсивности их использования. Например, можно отвести одну табличную область для таблиц, а другую — для индексов. В каждой базе данных есть табличная область SYSTEM, с которой связаны все системные объекты, например таблицы словаря данных. Доступность табличных областей может устанавливаться переводом в автономный или оперативный режим.

Для эффективного управления разграничением доступа к данным система Oracle поддерживает объект роль.

**Роль (ROLE)** — именованная совокупность привилегий, которые могут быть предоставлены пользователям или другим ролям. Oracle поддерживает несколько предопределенных ролей. Для систем, в которых количество пользователей и приложений велико, роли могут заметно облегчить разграничение доступа, например возможно динамически назначать роли для изменения набора привилегий пользователя при работе с различными при- ложениями. Также роли можно защищать паролем.

Специфичными для распределенных систем являются объекты Oracle: снимок и связь базы данных.

**Снимок (SNAPSHOT)** —локальная копия таблицы удаленной базы данных, которая используется либо для тиражирования (копирования) всей или части таблицы, либо для тиражирования результата запроса данных из нескольких таблиц. Снимки могут быть модифицируемыми или предназначенными только для чтения. Снимки только для чтения возможно периодически обновлять, отражая изменения основной таблицы. Изменения, сделанные в модифицируемом снимке, распространяются на основную таблицу и другие копии.

**Связь базы данных (DATABASE LINK)** — это объект базы данных, который позволяет обратиться к объектам удаленной базы данных. Имя связи базы данных можно рассматривать как ссылку на параметры механизма доступа к удаленной базе данных (имя узла, протокол и т. п.).

**Сегмент отката (ROLBACK SEGMENT)** — объект базы данных, предназначенный для обеспечения многопользовательской работы. В сегментах отката находятся обновляемые и удаляемые данные в пределах одной транзакции. При отмене изменений старая версия данных всегда доступна, так как находится в сегментах отката. В начале транзакции и в каждой контрольной точке текущие значения данных копируются в сегмент отката. Кроме того, сегменты отката используются при других операциях сервера. Размер и доступность сегментов отката в сильной степени влияют на производительность сервера баз данных и их настройка должна быть выполнена самым тщательным образом.

**Типы (TYPE) и коллекции типов** — новые виды объектов базы данных для Огас1е8, предназначенные для реализации объектных расширений.

**Каталог (DIRECTORY)** — объект, предназначенный для организации файлового ввода-вывода и работы с большими двоичными объектами.

**Профиль (PROFILE)** — объект, ограничивающий использование пользователем системных ресурсов, например процессорного времени или числа операций ввода-вывода.

**Библиотека (LIBRARY)** — объект базы данных, предназначенный для взаимодействия программ PL/SQL с модулями, написанными на других языках программирования.

К числу основных субъектов (активных сущностей) Oracle относятся: пользователь, процедура, функция, пакет и триггер.

**Пользователь (USER)** — сущность, обладающая возможностью создавать и использовать другие объекты Oracle, а также запрашивать выполнение функций сервера. К числу таких функций относится организация сессии, изменение состояния базы данных и др.

Следует отметить, что в некоторых других системах управления базами данных, например IBM DB2, объект базы данных «пользователь» отсутствует.

С пользователем Oracle связана схема (SCHEMA), которая является логическим набором объектов базы данных, таких, как таблицы, последовательности, хранимые программы, при- надлежащие этому пользователю. Схема имеет только одного пользователя-владельца, обладающего полным набором привилегий на создание, модификацию и удаление этих объектов. При создании пользователем первого объекта неявно создается соответствующая схема. При создании им других объектов они по умолчанию становятся частью этой схемы. Для просмотра объектов схемы текущего пользователя можно использовать представление словаря данных USEROBJECTS.

При массовом выполнении DDL-предложений можно создать несколько объектов и назначить для них привилегии за одну операцию, используя оператор CREATE SCHEMA. Оператор CREATE SCHEMA применяется тогда, когда требуется гарантировать успешное создание всех объектов и назначение привилегий за одну операцию. Если при создании объектов произошла ошибка, происходит возвращение к исходному состоянию.

Схема может содержать следующие объекты: кластеры, связи баз данных, триггеры, библиотеки внешних процедур, индексы, пакеты, последовательности, хранимые функции и процедуры, синонимы, таблицы, представления, снимки, объектные таблицы, объектные типы, объектные представления.

Объекты схемы могут состоять из других объектов, называемых подобъектами схемы. К ним относятся столбцы таблиц и представлений, секции таблиц, ограничения целостности, триггеры, пакетные процедуры и функции и другие элементы, хранимые в пакетах (курсоры, типы и т. п.). К объектам, не принадлежащим схеме, но хранимым в базе данных, относятся каталоги, профили, роли, сегменты отката, табличные области и пользователи.

Процедуры, функции, пакеты, тела пакетов и триггеры используются для программирования алгоритмов обработки данных.

**Процедура (PROCEDURE)** — это поименованный, структурированный набор конструкций языка PL/SQL, предназначенный для решения конкретной задачи.

**Функция (FUNCTION)** —это поименованный, структурированный набор конструкций языка PL/SQL, предназначенный для решения конкретной задачи и возвращающий значение.

**Пакет (PACKAGE)** —это поименованный, структурированный набор переменных, процедур и функций и других объектов, связанных функциональным замыслом. Пакет состоит из двух самостоятельных частей: заголовка и тела. Заголовок содержит описание переменных, констант, типов, процедур, функций и других конструкций языка PL/SQL. Тело пакета содержит реализацию алгоритмов процедур и функций и хранится отдельно. Например, Oracle предоставляет стандартный пакет UTL\_FILE, который содержит процедуры и функции, предназначенные для организации файлового ввода-вывода из программ на языке PL/SQL.

**Триггер (TRIGGER)** — это хранимая процедура, которая автоматически выполняется тогда, когда происходит связанное с триггером событие. Обычно события связаны с выполнением операторов вставки, модификации и удаления данных. С помощью триггеров можно реализовать правила динамической проверки целостности данных и дополнительного контроля доступа.