Защита данных в базах данных: защита от несанкционированного доступа

"Стыдно не уметь защищать себя рукою, но ещё более стыдно не уметь защищать себя словом".
Аристотель, древнегреческий философ

Защита от несанкционированного доступа

Под функцией секретности данных понимается защита данных от преднамеренного искажения и/или доступа пользователей или посторонних лиц. Для этого вся информация делится на общедоступные данные и конфиденциальные, доступ к которым разрешен только для отдельных групп лиц. Решение этого вопроса относится к компетенции юридических органов или администрации предприятия, для которого создаётся БД, и является внешней функцией по отношению к БД.

Общий принцип управления доступом к базе данных такой: СУБД не должна разрешать пользователю выполнение какой-либо операции над данными, если он не получил на это права. Санкционирование доступа к данным осуществляется администратором БД.

В обязанности администратора БД входит:

- назначение отдельным группам пользователей прав доступа (привилегий) к отдельным группам данных в соответствии с правилами ПО;
- организация системы контроля доступа к данным;
- тестирование вновь создаваемых средств защиты данных;
- периодическое проведение проверок правильности работы системы защиты, исследование и предотвращение сбоев в её работе.

Защита от несанкционированного доступа

В стандарте SQL/3 предусмотрены возможности контроля доступа к разным объектам базы данных, в том числе к следующим объектам:

Вид защиты и соответствующее действие	Название привилегии	Применимо к следующим объектам
Просмотр	SELECT	Таблицы, столбцы, подпрограммы, вызываемые из SQL
Вставка	INSERT	Таблицы, столбцы
Модификация	UPDATE	Таблицы, столбцы
Удаление	DELETE	Таблицы
Ссылка	REFERENCES	Таблицы, столбцы
Использование	USAGE	Домены, определенные пользователями типы, наборы символов, порядки сортировки символов, преобразования
Инициирование	TRIGGER	Таблицы
Выполнение	EXECUTE	Подпрограммы, вызываемые из SQL
Подтипизация	UNDER	Структурные типы

Защита от несанкционированного доступа

Каждый пользователь имеет свой паспорт, который содержит:

- его идентификатор (authorization identifier – authID, или login);
- имя процедуры подтверждения подлинности;
- [пароль];
- перечень разрешённых операций.

В качестве процедуры аутентификации обычно применяется парольная идентификация.

Парольная идентификация заключается в присвоении каждому пользователю двух параметров: имени (login) и пароля (password).

При задании пароля желательно соблюдать следующие требования:

- длина пароля должна быть не менее 6-и символов;
- пароль должен содержать комбинацию букв и цифр или специальных знаков, пароль не может содержать пробелы;
- пароли должны часто меняться.

Для контроля выполнения этих требований обычно применяются специальные программы.

Управление доступом

Для подключения к базе данных у пользователя должна быть учетная запись. Параметры учетной записи, права и привилегии определяют возможности пользователя в пределах базы данных.

Команда создания пользователей Oracle использует следующий синтаксис:

```
CREATE USER имя_пользователя IDENTIFIED { BY пароль |
                        EXTERNALLY }
    [ DEFAULT TABLESPACE имя_табличной_области1 ]
    [ TEMPORARY TABLESPACE имя_табличной_области2 ]
    [ QUOTA {число_единиц [{ K | M }] | UNLIMITED }
          ON имя _табличной области1 ]
    [ QUOTA {число_единиц [{ K | M }] | UNLIMITED }
          ON имя _табличной области2  ... ]
    [ PROFILE имя_профиля ];
```

Нельзя не указывать табличную область по умолчанию, иначе ею будет назначена табличная область SYSTEM!

Управление доступом

Имя пользователя

Имя пользователя в пределах базы данных должно быть уникальным. Длина не должна превышать 30 символов. Имя может состоять из латинских букв, цифр, знака доллар ($) и знака подчеркивания (_). Имя пользователя не может быть зарезервированным словом.

Если требуется создать пользователя с какими либо спец символами в имени, то это можно обойти заключив имя в двойные кавычки.

До версии Oracle Database 11g регистр имени не учитывался, и оно автоматически переводилось в верхний регистр. Начиная с 11 версии, появилась возможность использования регистрозависимых учетных записей.

После того как учетная запись создана изменить ее имя нельзя.

Для изменения придется создать новую, с требуемым именем и удалить старую. Огромная неприятность в случае удаления учетной записи, это то, что вместе с ней удаляются и все объекты, принадлежащие этому пользователю.

Управление доступом

Метод аутентификации

Различаются несколько методов аутентификации пользователей в Oracle: парольная, внешняя и глобальная.

Парольная аутентификация

При попытке пользователя подключится к базе данных с использованием парольной аутентификации, база данных проверят имя пользователя и пароль на совпадение с сохраненными в ней данными. В случае успеха пропускает, в противном случае в доступе отказывается.

Пароль пользователя хранится в словаре данных в зашифрованном виде.

Пример:

```
CREATE USER ALL_ORACLE IDENTIFIED BY qwerty;
```

Управление доступом

Метод аутентификации

Внешняя аутентификация

При внешней аутентификации, при попытке пользователя подключится к базе данных, она проверяет учетную запись пользователя, и доверяет операционной системе. Т.е. если операционная система пропустила пользователя, то база данных принимает его.

Для пользователей авторизуемых операционной системой база данных не хранит пароли и не проверяет их корректность. Когда в Oracle появилась такая возможность, в Oracle 6, такие учетные записи имели префикс OPS$. Начиная с 6 версии таких пользователей можно настраивать указав OS_AUTHENT_PREFIX при инициализации или в SPFILE файле. Например, создадим пользователя ALL_ORACLE_EXT:

```
CREATE USER OPS$ALL_ORACLE_EXT
        IDENTIFIED EXTERNALLY;
```

Часто учетные записи с внешней аутентификацией используются для исполнения административных скриптов, которые могут быть прочитаны посторонними людьми, и хранение пароля в открытом виде крайне опасно.

Управление доступом

Метод аутентификации

Глобальная аутентификация

Одним из развивающихся стандартов глобальной аутентификации, это использование LDAPсерверов (например, Oracle Internet Directory). При попытке пользователя подключится к базе данных с использованием глобальной аутентификации, база данных проверят имя пользователя и корректность аутентификационной информации в LDAP директории.

Продвинутыми опциями безопасности являются биометрические параметры, сертификаты X.509, RADIUS и Kerberos.

Для таких учетных записей пароли в базе данных не сохраняются, а аутентификация проходит посредством продвинутых механизмов безопасности. Ниже представлен пример создания такого пользователя:

```
CREATE USER ALL_ORACLE_GLOBAL IDENTIFIED GLOBALLY
    AS 'CN=global, OU=tier1, O=security, C=US';
```

Существует два механизма глобальной аутентификации:

1. Для пользователя учетная запись может быть заведена как в базе данных так и в глобальной директории. Пользователи подключаются с учетной записью, и такое же имя существует в директории.
2. Учетная запись определена только в глобальной директории, база данных знает обо всех пользователях директории, но они все заходят в базу данных под одной общей учетной записью в базе данных.

Управление доступом

Табличное пространство по умолчанию и ограничения

Каждому пользователю назначается табличное пространство по умолчанию. Указать табличное пространство по умолчанию можно либо при создании, либо при модификации учетной записи:

1) CREATE USER ALL_ORACLE IDENTIFIED BY qwerty
DEFAULT TABLESPACE USERS;
2) ALTER USER ALL_ORACLE DEFAULT TABLESPACE USERS;

Для изменения табличного пространства по умолчанию в базе данных используется предложение ALTER DATABASE:

ALTER DATABASE DEFAULT TABLESPACE users;

В случае создания объектов пользователем, если для него явно не указано табличное пространство по умолчанию, то будет использоваться табличное пространство определенное по умолчанию для базы данных.

После создания базы данных никогда не оставляйте по умолчанию табличное пространство SYSTEM. Либо смените во время создания, либо после создания базы данных укажите другое табличное пространство по умолчанию.

Управление доступом

Табличное пространство по умолчанию и ограничения

Квота на табличное пространство – это объем пространства который может использовать пользователь. В пределах указанных ограничений он может создавать объекты, хранить данные и т.д. Как только предел будет достигнут, пользователь не сможет ничего сохранить.

Изменить квоту можно в любое время.

Примеры назначения квоты на табличные пространства для пользователя ALL_ORACLE:

1) ALTER USER ALL_ORACLE QUOTA 100M ON SYSTEM;
2) ALTER USER ALL_ORACLE QUOTA UNLIMITED ON USERS;
3) ALTER USER ALL_ORACLE QUOTA UNLIMITED ON EXAMPLE;

Получить данные о квотах на табличные пространства и табличных пространствах пользователя можно, выполнив запросы:

```
SELECT USERNAME, DEFAULT_TABLESPACE, TEMPORARY_TABLESPACE
    FROM DBA_USERS
    WHERE USERNAME = 'TEST';
```

Управление доступом

Временное табличное пространство

Каждому пользователю назначается временное табличное пространство (тип TEMPORARY), в котором база данных хранит временные сегменты.

Временные сегменты:

- создаются во время операций: ORDER BY, GROUP BY, SELECT DISTINCT, MERGE JOIN или CREATE INDEX;
- используются при использовании временных таблиц.

Пример указания временного табличного пространства пользователю:

```
CREATE USER ALL_ORACLE IDENTIFIED BY qwerty
    DEFAULT TABLESPACE USERS
    TEMPORARY TABLESPACE TEMP;
```

Или в предложении ALTER USER:

```
    ALTER USER ALL_ORACLE TEMPORARY TABLESPACE TEMP;
```

Для изменения временного табличного пространства в базе данных используется конструкция ALTER DATABASE:

```
    ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temp;
```

Пользователю не требуется квота на временное табличное пространство.

Управление доступом

Назначение профиля пользователю

Профиль содержит:

1. информацию об ограничениях на использование ресурсов;
2. правила управления паролем.

Профиль по умолчанию так и называется – default. Для явного назначения профиля пользователю в предложение CREATE USER или ALTER USER добавляется ключевое слово PROFILE и указывается имя профиля. Например:

```
CREATE USER ALL_ORACLE IDENTIFIED BY qwerty
    DEFAULT TABLESPACE USERS
    TEMPORARY TABLESPACE TEMP
    PROFILE resource_profile;
```

или

```
    ALTER USER ALL_ORACLE PROFILE resource_profile;
```

Управление доступом

Статусы учетной записи

У каждой учетной записи есть свой статус. Узнать его можно так:

```
SELECT  USERNAME,  ACCOUNT_STATUS
        FROM  DBA_USERS;
```

Статусы:

OPEN – учетная запись доступна для использования.

LOCKED – учетная запись заблокирована DBA.

EXPIRED – истекло время действия пароля.

EXPIRED & LOCKED – учетная запись не только заблокирована, но и истекло время действия пароля.

EXPIRED (GRACE) – сигнализирует о действии дополнительного времени на действие пароля.

LOCKED (TIMED) – учетная запись заблокирована после нескольких неудачных попыток подключится к базе данных.

EXPIRED & LOCKED (TIMED)

EXPIRED (GRACE) & LOCKED

EXPIRED (GRACE) & LOCKED (TIMED)

Для блокировки и разблокировки учетной записи используются команды:

```
ALTER  USER  username  ACCOUNT  LOCK;
ALTER  USER  username  ACCOUNT  UNLOCK;
```

Предоставление прав доступа

Предоставление прав доступа (привилегий) в системах, поддерживающих язык SQL, осуществляется с помощью двух команд:

1) Объектные привилегии:

GRANT { <список привилегий> | ALL [PRIVILEGES] } ON <имя объекта>
TO {<список пользователей> | PUBLIC}
[WITH GRANT OPTION];

2) Системные привилегии (там, где они поддерживаются):

GRANT <список привилегий>
TO {<список пользователей> | PUBLIC }
[WITH GRANT OPTION];

где <список привилегий> – набор прав, которые необходимо предоставить, или ALL PRIVILEGES – все права на данный объект;

<имя объекта> – имя объекта БД, к которому предоставляется доступ;

<список пользователей> – перечень пользователей (или *ролей*), которым будут предоставлены указанные права;

PUBLIC – предопределённый пользователь, привилегии которого доступны всем пользователям БД.

WITH GRANT OPTION – возможность передавать права доступа другим пользователям (но не ролям!).

Отмена привилегий

Отмена прав доступа (привилегий) в системах, поддерживающих язык SQL, осуществляется с помощью команды **REVOKE:**

```
REVOKE [GRANT OPTION FOR]
{ <список привилегий> | ALL PRIVILEGES }
ON <имя объекта>
FROM {<список пользователей> | PUBLIC }
{ RESTRICT | CASCADE };
```

где

[GRANT OPTION FOR] – отмена права передачи привилегий;

CASCADE – при отмене привилегий у пользователя отменяются все привилегии, которые он передавал другим пользователям;

RESTRICT – если при отмене привилегий у пользователя необходимо отменить переданные другим пользователям привилегии, то операция завершается с ошибкой.

Если отменяются системные привилегии, то имя объекта не указывается.

Объектные привилегии Oracle

Привилегия	Разрешение на
ALL	все действия с объектом
ALL PRIVILEGES	то же, что ALL
ALTER	изменение определения объекта
DELETE	удаление строк из таблицы, представления
EXECUTE	выполнение объекта, на обращение к его переменным
INDEX	создание индексов по таблице
INSERT	вставку строк в таблицу, представление
REFERENCES	создание ограничений, которые ссылаются на таблицу
SELECT	выборку строк из таблицы, представления, моментального снимка или последовательности
UPDATE	изменение строк в таблице, представлении

Если все объектные привилегии назначены с помощью обозначения ALL, индивидуальные привилегии по-прежнему могут быть отозваны.

Объектные привилегии Oracle

Объектная привилегия	Таблицы	Представ-ления	Последова-тельности	Процедуры Функции Пакеты	Снапшоты
ALTER	+		+		
DELETE	+	+			
EXECUTE				+	
INDEX *	+				
INSERT	+	+			
REFERENCES *	+				
SELECT	+	+	+		+
UPDATE	+	+			

* - привилегия не может быть назначена роли

Предложения SQL, допускаемые объектными привилегиями базы данных

Привилегия	Допустимые предложения SQL
ALTER	ALTER объект (таблица или последовательность)
DELETE	DELETE FROM объект (таблица или представление)
EXECUTE	EXECUTE объект (процедура или функция), обращения к общим переменным в пакете
INDEX	CREATE INDEX ON объект (только таблицы)
NSERT	INSERT INTO объект (таблица или представление)
REFERENCES	Предложение CREATE или ALTER TABLE, определяющее ограничение целостности FOREIGN KEY по объекту (только таблицы)
SELECT	SELECT ... FROM объект (таблица, представление, снимок), предложения SQL, использующие последовательность
UPDATE	UPDATE объект (таблица или представление)

Назначение объектных привилегий

Примеры:

GRANT ALL ON tab to user1, user2;

GRANT SELECT, UPDATE, DELETE ON tab to user1, user2;

GRANT INSERT(field1, filed2), UPDATE(field1, filed2) ON tab to user3;

Прежде чем назначать привилегию INSERT, специфичную для столбцов, надо рассмотреть возможные побочные эффекты; если таблица имеет один или несколько столбцов, объявленных как NOT NULL, то выборочная привилегия INSERT, не включающая этих столбцов, не имеет смысла, так как она может не позволить получившему ее пользователю вставить в таблицу ни одной строки.

Чтобы предотвратить проблемы, необходимо проверить, чтобы каждый столбец с ограничением NOT NULL либо входил в привилегию INSERT, либо имел непустое умалчиваемое значение; если это не учесть, будет возвращаться ошибка, и строки не смогут быть вставлены.

Продвижение объектных привилегий

Опция GRANT OPTION дает следующие дополнительные возможности:
* Пользователь может назначать эту объектную привилегию любому другому пользователю или роли в базе данных (с опцией GRANT OPTION или без таковой).
* Если пользователь получил объектные привилегии для таблицы с опцией GRANT OPTION, и он имеет системную привилегию CREATE VIEW или CREATE ANY VIEW, то он может создавать представления по этой таблице, и назначать соответствующие привилегии по этому представлению любому пользователю или роли в базе данных.

Пользователь, схема которого содержит объект, автоматически имеет все ассоциированные объектные привилегии для этого объекта с опцией GRANT OPTION.

Опция GRANT OPTION недопустима при назначении объектной привилегии РОЛИ. ORACLE предотвращает распространение объектных привилегий через роли, так что получатели роли не могут дальше продвигать свои объектные привилегии, полученные через роли.
Чтобы назначить кому-либо объектную привилегию, пользователь должен либо владеть соответствующим объектом, либо иметь для объектных привилегий опцию GRANT OPTION.

Отзыв объектных привилегий

Пример отзыва объектных привилегий:

```
REVOKE select, insert ON emp FROM jward, tsmith;
```

Можно также отозвать все привилегии по таблице DEPT (даже если была назначена лишь одна привилегию), назначенные роли HUMAN_RESOURCES, введя следующее предложение:

```
REVOKE ALL ON dept FROM human_resources;
```

Данное предложение отзовет лишь те привилегии, на которые имеет соответствующие полномочия пользователь, запускающий это предложение, но не все привилегии, которые были назначены другими.

Нельзя выборочно отозвать опцию GRANT OPTION, не отзывая привилегию на объект; чтобы сделать это, следует отозвать объектную привилегию и заново назначить ее без опции GRANT OPTION.

Пользователь не может отозвать объектную привилегию у самого себя.

Отзыв объектных привилегий

Выборочные привилегии SELECT, UPDATE и REFERENCES по отдельным столбцам таблиц и представлений не могут выборочно отзываться аналогичным предложением REVOKE. Вместо этого следует сначала отозвать объектную привилегию по всем столбцам таблицы или представления, а затем вновь выборочно назначить привилегии по тем столбцам, которые должны остаться.

Например, предположим, что роли HUMAN_RESOURCES была назначена привилегия UPDATE по столбцам DEPTNO и DNAME таблицы DEPT. Чтобы отозвать привилегию UPDATE по столбцу DEPTNO и оставить ее по столбцу DNAME, нужно ввести следующие два предложения:

```
REVOKE UPDATE ON dept FROM human_resources;
GRANT UPDATE (dname) ON dept TO human_resources;
```

Если пользователь, получивший **привилегию REFERENCES**, использовал эту привилегию для создания ограничения внешнего ключа (которое существует в данный момент), то пользователь, назначавший эту привилегию, может отозвать ее лишь каскадно:

```
REVOKE REFERENCES ON dept FROM jward CASCADE CONSTRAINTS;
```

При этом все ограничения внешних ключей, использующие отзываемую привилегию REFERENCES, удаляются.

Отзыв объектных привилегий

* При отзыве объектной привилегии DML могут быть затронуты определения объектов, зависящих от этой объектной привилегии DML. Например, пусть процедура TEST включает предложение SQL, которое опрашивает таблицу EMP. Если привилегия SELECT по таблице EMP будет отозвана у владельца процедуры TEST, то процедура перестанет успешно выполняться.
* Определения объектов, требующие объектных привилегий DDL ALTER и INDEX, не затрагиваются при отзыве этих объектных привилегий ALTER и INDEX. Например, при отзыве привилегии INDEX у пользователя, создавшего индекс по не своей таблице, этот индекс останется.
* Назначения объектных привилегий, которые были распространены с помощью GRANT OPTION, автоматически отзываются при отзыве объектной привилегии у того, кто имел опцию GRANT OPTION. Например, предположим, что пользователь USER1 получил объектную привилегию SELECT с опцией GRANT OPTION, и назначил привилегию SELECT по таблице EMP пользователю USER2. Впоследствии привилегия SELECT у пользователя USER1 отзывается. Этот отзыв каскадно распространяется и на пользователя USER2. Все объекты, которые зависели от отозванных привилегий SELECT пользователей USER1 и USER2, будут также затронуты.

Системные привилегии Oracle

Системная привилегия	Разрешение на
ANALYZE ANY	анализ любых таблиц, индексов и кластеров в любых схемах с помощью команды ANALYZE
AUDIT ANY	аудит любых объектов в любых схемах
ALTER DATABASE	изменение базы данных
SELECT ANY DICTIONARY	чтение словаря-справочника данных
ALTER / AUDIT SYSTEM	изменение системы/аудит системных событий
CREATE / ALTER / DROP [ANY] CLUSTER	создание/изменение/удаление кластеров [во всех схемах]
TRUNCATE ANY	опустошение любых таблиц и кластеров
CREATE / DROP [PUBLIC] DATABASE LINK	создание/удаление связей базы данных
CREATE / ALTER / DROP [ANY] INDEX	создание/изменение/удаление индексов [во всех схемах]
CREATE / ALTER / DROP [ANY] PROCEDURE	создание/изменение/удаление [любой] процедуры, функции, [тела] пакета
EXECUTE ANY PROCEDURE	выполнение любой процедуры, функции, пакета

Системные привилегии Oracle

Системная привилегия	Разрешение на
CREATE / ALTER / DROP PROFILE	создание/изменение/удаление профилей
CREATE / ALTER / DROP [ANY] ROLE	создание/изменение/удаление [любых] ролей
GRANT ANY ROLE	предоставление любых ролей в базе данных
GRANT ANY PRIVILEGE	предоставление любых системных привилегий
CREATE / ALTER / DROP ROLLBACK SEGMENT	создание/изменение/удаление сегментов отката
CREATE / ALTER SESSION	соединение/изменение параметров сессии
RESTRICTED SESSION	соединение при установке STARTUP RESTRICT
CREATE / ALTER / DROP [ANY] SEQUENCE	создание/изменение/удаление [любых] последовательностей
SELECT ANY SEQUENCE	просмотр из любых последовательностей
CREATE / ALTER / DROP [ANY] SNAPSHOT	создание/изменение/удаление [любых] синонимов
CREATE / DROP [PUBLIC] / [ANY] SYNONYM	создание/изменение/удаление [общих]/[любых] синонимов

Системные привилегии Oracle

Системная привилегия	Разрешение на
CREATE / ALTER / DROP [ANY] TABLE	создание/изменение/удаление [любых] таблиц
BACKUP ANY TABLE	экспорт записей из любых таблиц
COMMENT ANY TABLE	комментирование таблиц, представлений, столбцов
SELECT / UPDATE ANY TABLE	просмотр/обновление записей любых таблиц
INSERT / DELETE ANY TABLE	вставка/удаление записей любых таблиц
LOCK ANY TABLE	блокирование любых таблиц в любых схемах
CREATE / ALTER / DROP TABLESPACE	создание/изменение/удаление табличных областей
MANAGE TABLESPACE	Управление табличными областями (ТО) (ONLINE/OFFLINE/BACKUP)
UNLIMITED TABLESPACE	неограниченную квоту памяти во всех ТО
CREATE / ALTER / DROP [ANY] TRIGGER	создание/изменение/удаление [любых] триггеров
CREATE / ALTER / DROP USER	создание/изменение/удаление пользователей
BECOME USER	становиться другим пользователем для импорта
CREATE / ALTER / DROP [ANY] VIEW	создание/изменение/удаление [любых] представлений

Отзыв системных привилегий Oracle

При отзыве системной привилегии, относящейся к операции DDL, каскадных эффектов не бывает, независимо от того, была ли эта привилегия назначена с опцией ADMIN OPTION. Например:

1. Администратор назначает пользователю JWARD системную привилегию CREATE TABLE с опцией ADMIN OPTION.
2. JWARD создает таблицу.
3. JWARD назначает системную привилегию CREATE TABLE пользователю TSMITH.
4. TSMITH создает таблицу.
5. Администратор отзывает у пользователя JWARD системную привилегию CREATE TABLE.
6. Таблица пользователя JWARD продолжает существовать. TSMITH по-прежнему имеет привилегию CREATE TABLE, и его таблица существует.

Каскадные эффекты можно наблюдать при отзыве системной привилегии, связанной с операцией DML. Например, если пользователю назначена привилегия SELECT ANY TABLE, и этот пользователь создал какие-либо процедуры, то все процедуры в схеме этого пользователя должны быть заново откомпилированы после отзыва этой привилегии, прежде чем их можно будет использовать вновь.

Назначение/отзыв привилегий для PUBLIC

Привилегии и роли можно также назначать и отзывать у группы пользователей PUBLIC. Поскольку группа PUBLIC доступна каждому пользователю базы данных, все привилегии и роли, назначенные PUBLIC, доступны каждому пользователю базы данных.
Администраторы защиты и пользователи должны назначать группе PUBLIC лишь те привилегии и роли, которые действительно необходимы каждому пользователю. Эта рекомендация согласуется с общим правилом, согласно которому в любой момент каждый пользователь базы данных должен иметь лишь те привилегии, которые требуются ему для успешного выполнения текущей задачи.
Отзыв прав у PUBLIC может повлечь за собой значительные каскадные эффекты, в зависимости от того, какая привилегия отзывается. Если у PUBLIC отзывается любая привилегия, связанная с операцией DML (например, SELECT ANY TABLE, UPDATE ON, и т.п.), то все процедуры в базе данных (включая функции и пакеты) должны быть заново АВТОРИЗОВАНЫ, прежде чем их можно будет использовать снова. Поэтому будьте осторожны, назначая группе PUBLIC привилегии, связанные с операциями DML.

Операции, влияющие на состояние объекта

Операция	Результирующее состояние объекта	Результирующее состояние зависимых объектов
CREATE таблица, посл., синоним	VALID, если нет ошибок	Без изменений*
ALTER таблица [ADD | MODIFY] столбец RENAME табл., посл., синоним, view	VALID, если нет ошибок	INVALID
DROP таблица, посл., view, процедура, функция, пакет	Никакого; объект удален	INVALID
CREATE view, процедура**	VALID, если нет ошибок	без изменений
CREATE OR REPLACE view, процедура	VALID, если нет ошибок	INVALID
REVOKE объектная привилегия ON объект FROM пользователь ***	Без изменений	INVALID для всех зависимых объектов этого пользователя
REVOKE объектная привилегия ON объект FROM PUBLIC ***	Без изменений	INVALID для всех зависимых объектов в базе данных
REVOKE системная привилегия FROM пользователь ****	Без изменений	INVALID для всех объектов пользователя
REVOKE системная привилегия FROM PUBLIC****	Без изменений	INVALID для всех объектов в базе данных

* Может вызывать недействительность зависимых объектов, если объект не существовал ранее.
** Независимые процедуры и функции, пакеты и триггеры.
*** Только объектные привилегии DML, включая SELECT, INSERT, UPDATE, DELETE и EXECUTE; приведение в действительное состояние не требует перекомпиляции.
**** Только системные привилегии DML, включая SELECT / INSERT / UPDATE / DELETE ANY TABLE и EXECUTE ANY PROCEDURE; приведение в действительное состояние не требует перекомпиляции.

Роли

Роль – это совокупность привилегий, предоставляемых пользователю и/или другим ролям. Создается так:

CREATE ROLE rolename [IDENTIFIED BY pswd];

В любой базе данных ORACLE автоматически определены роли CONNECT, RESOURCE, DBA, EXP_FULL_DATABASE и IMP_FULL_DATABASE. Эти роли предоставляются для совместимости с более ранними версиями ORACLE. Их можно модифицировать точно так же, как и любую другую роль в базе данных ORACLE.

Примеры:

create role stud;

grant connect, create session, create table, select any table, create any view, create procedure, execute any procedure, create trigger, create sequence, create any synonym, select any dictionary, audit any
to stud;

create user user1 identified by user1
default tablespace users quota unlimited on users;

grant stud to user1;

grant select on system.persons to PUBLIC;

Определение разрешенных ролей

Роли, которые назначены на текущий сеанс, перечислены в представлении словаря данных SESSION_ROLES:

SELECT ROLE FROM SESSION_ROLES;

Этот список содержит роли назначенные пользователю, от имени которого запускается запрос, пользователю PUBLIC и роли, которые наследуются от других ролей. Например, роль DBA включает роль SCHEDULER_ADMIN, которая содержит системные привилегии (например, CREATE ANY JOB).

Разрешить роль можно так:

SET ROLE ALL_ORACLE_DBA IDENTIFIED BY "all_oracle.ru", ALL_ORACLE;

Разрешения всех ролей, за исключением ALL_ORACLE_DBA:

SET ROLE ALL EXCEPT ALL_ORACLE_DBA;

В пределах сеанса можно запретить только роли, назначенные явно, или роли, назначенные пользователю PUBLIC:

SET ROLE NONE; - запрещение всех ролей

SET ROLE ALL EXCEPT [роль1, роль2…]; - запрещения ролей за исключением каких-либо

Нельзя запретить роли, которые наследуются от других ролей, без запрещения родительской роли.

Назначение ролей по умолчанию и явно

Сразу после создания пользовательского аккаунта, для него назначаются все роли по умолчанию (DEFAULT ROLE ALL).

Если у пользователя есть роль по умолчанию (DEFAULT ROLE), то он может ей пользоваться только в случае, когда она ему непосредственно присвоена до этого.

Опции команды изменения аккаунта:

DEFAULT ROLE имя_роли – роль, назначаемая пользователю сразу после открытия сессии (без необходимости вводить пароль такой роли);

DEFAULT ROLE ALL – пользователю назначаются все, присвоенные ему роли, за исключением тех, которые указаны после предложения EXCEPT;

DEFAULT ROLE NONE – все роли, назначенные пользователю, будут отключены после открытия сессии.

Если какие либо роли исключаются из DEFAULT, то пользователь не сможет воспользоваться соответствующими правами, пока не включит роль командой SET ROLE ИМЯ_РОЛИ.

Изменение аккаунта пользователя

```
ALTER USER имя_пользователя
IDENTIFIED { BY пароль | EXTERNALLY | GLOBALLY AS `CN=user' }
[ DEFAULT TABLESPACE имя_табличного_пространства ]
[ TEMPORARY TABLESPACE имя_табличного_пространства ]
[ QUOTA {число [{ K | M }] | UNLIMITED ] ON имя_табличного_пространства ]
[, QUOTA {число [{ K | M }] | UNLIMITED] ON имя_табличного_пространства ]
[ PROFILE наименования_профиля ]
[ PASSWORD EXPIRE ]
[ ACCOUNT LOCK или ACCOUNT UNLOCK ]
[ DEFAULT ROLE { имя_роли [, имя_роли ] |
                    ALL [ EXCEPT имя_роли [, имя_роли ] ] ] | NONE ] ;
```

* Все назначения/отзывы привилегий (системных и объектных) кому угодно (пользователям, ролям, PUBLIC) наблюдаются немедленно.
* Все назначения/отзывы ролей кому угодно (пользователям, ролям, PUBLIC) наблюдаются лишь после того, как текущая сессия пользователя выдает предложение SET ROLE для повторного включения роли, или при создании новой сессии пользователя.

Профили

Профиль (PROFILE) – это набор ограничений на количество системных ресурсов, которые данный пользователь может занимать:

- CPU/session – время центрального процессора (ЦП), которое пользователь может занимать в течение сессии, в секундах. Возможные значения: DEFAULT, UNLIMITED, 1'000, 6'000, 36'000.
- CPU/call – время центрального процессора (ЦП), которое пользователь может занимать в течение одного обращения к БД, в секундах. Под обращением подразумевается: разбор предложения (parse), выполнение (execute) или извлечение следующей строки (fetch). Возможные значения: DEFAULT, UNLIMITED, 1'000, 6'000, 36'000.
- Connect time – максимальная продолжительность сессии, в минутах. Возможные значения: DEFAULT, UNLIMITED, 30, 60, 120.
- Idle time – максимально допустимое время простоя (неактивности) пользователя в течение сессии), в минутах. Возможные значения: DEFAULT, UNLIMITED, 1, 15, 60.
- Concurrent sessions – максимальное количество параллельных сессий от имени одного пользователя. Возможные значения: DEFAULT, UNLIMITED, 1, 2, 10.

Профили

- Reads/session – общее количество блоков данных, считанных за одну сессию. Измеряется в блоках. Возможные значения: DEFAULT, UNLIMITED, 1'000, 5'000, 10'000.
- Reads/call – максимальное количество блоков данных, считанных за один вызов (parse, execute, fetch). Измеряется в блоках. Возможные значения: DEFAULT, UNLIMITED, 1'000, 5'000, 10'000.
- Private SGA – максимальный объём частной SQL-области, выделенной пользователю в SGA (system global area, глобальная системная область оперативной памяти). Используется только для режима многоканального сервера (Multi-Tread server, MTS). Измеряется в килобайтах. Возможные значения: DEFAULT, UNLIMITED, 4, 16, 256.
- Composite limit – общая стоимость ресурсов на сессию. Рассчитывается как взвешенная сумма времени ЦП на сессию, времени подключения, количества физических чтений и объёма частной SQL-области. Возможные значения: DEFAULT, UNLIMITED, 1'000'000, 5'000'000, 10'000'000. Значение UNLIMITED означает отсутствие ограничения, а DEFAULT – значение этого ограничения, взятое из профиля с именем DEFAULT, который назначается любому пользователю автоматически.

Профили

- Reads/session – общее количество блоков данных, считанных за одну сессию. Измеряется в блоках. Возможные значения: DEFAULT, UNLIMITED, 1'000, 5'000, 10'000.
- Reads/call – максимальное количество блоков данных, считанных за один вызов (parse, execute, fetch). Измеряется в блоках. Возможные значения: DEFAULT, UNLIMITED, 1'000, 5'000, 10'000.
- Private SGA – максимальный объём частной SQL-области, выделенной пользователю в SGA (system global area, глобальная системная область оперативной памяти). Используется только для режима многоканального сервера (Multi-Tread server, MTS). Измеряется в килобайтах. Возможные значения: DEFAULT, UNLIMITED, 4, 16, 256.
- Composite limit – общая стоимость ресурсов на сессию. Рассчитывается как взвешенная сумма времени ЦП на сессию, времени подключения, количества физических чтений и объёма частной SQL-области. Возможные значения: DEFAULT, UNLIMITED, 1'000'000, 5'000'000, 10'000'000. Значение UNLIMITED означает отсутствие ограничения, а DEFAULT – значение этого ограничения, взятое из профиля с именем DEFAULT, который назначается любому пользователю автоматически.

Удаление ролей, профилей, пользователей

1) DROP ROLE имя_роли;

Все роли, косвенно назначавшиеся через удаленную роль, также будут удалены из затрагиваемых доменов защиты. Удаление роли автоматически удаляет ее из списков умалчиваемых ролей всех пользователей. Поскольку создание объектов не зависит от привилегий, полученных через роль, таблицы и другие объекты не удаляются при удалении роли.

2) DROP PROFILE имя_профиля;

Профиль не может быть удален, пока он назначен пользователям. Предварительно им требуется назначить другой профиль, и затем удалить требуемый, выполнив:

Как альтернативу можно использовать:

DROP PROFILE имя_профиля CASCADE;

Эта конструкция автоматически всем пользователям, которым назначен профиль с именем имя_профиля, будет назначен профиль DEFAULT.

3) DROP USER имя_пользователя [CASCADE];

Удаление пользователя неявно удаляет все объекты принадлежащие ему, но не роли, назначенные пользователю.

Использование представлений для определения прав доступа

Создать представление "Сотрудники 2-го отдела" (для предоставления полного доступа к данным о сотрудниках 2-го отдела начальнику этого отдела):

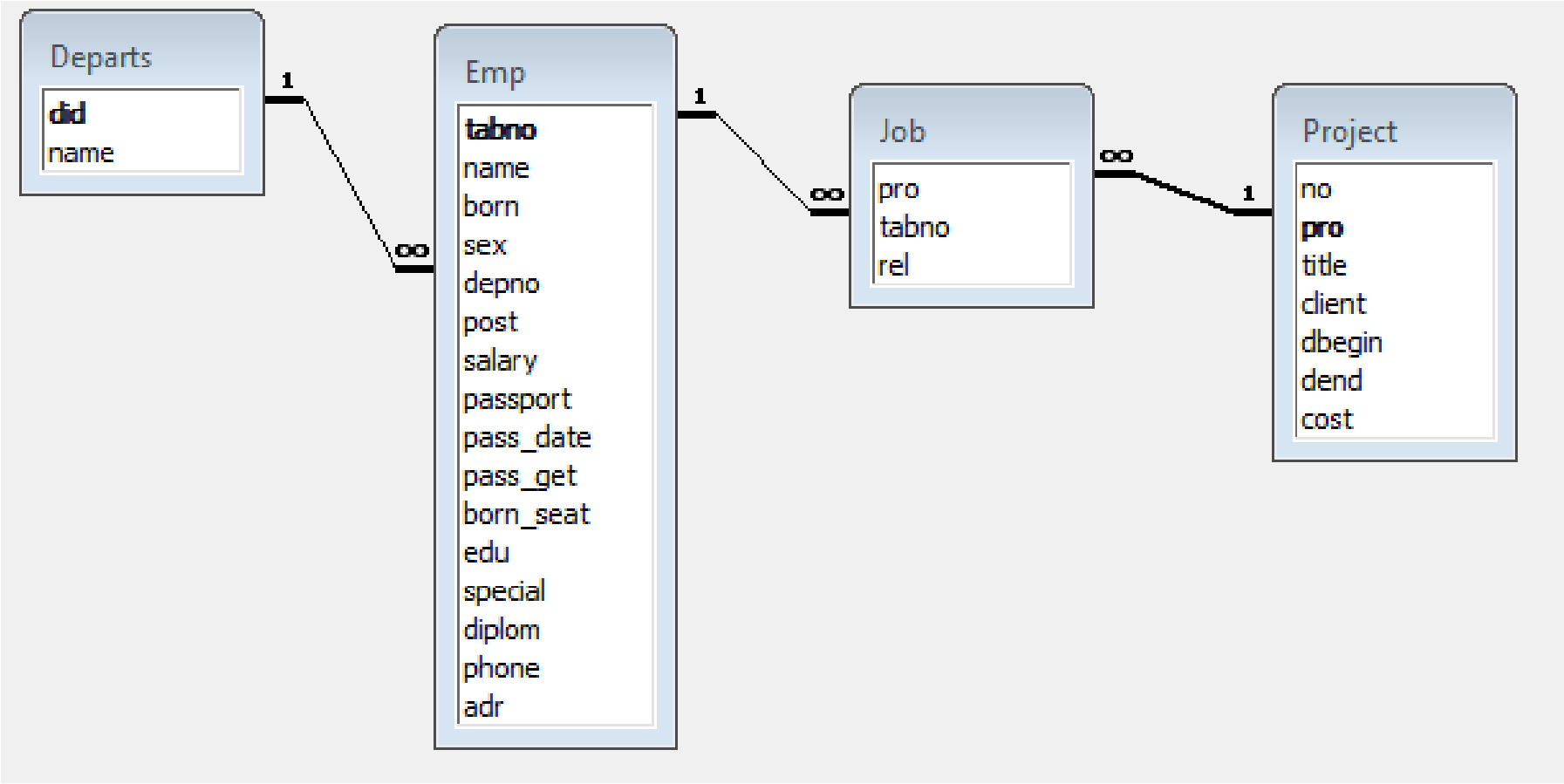
```
CREATE  VIEW   emp2
       AS  SELECT  *
       FROM  emp
       WHERE  depno = 2
       WITH  CHECK  OPTION;
GRANT  ALL  ON  emp2  TO  chief2;
```

Создать представление "Сотрудники" (без данных о зарплате, для сокрытия конфиденциальной информации):

```
CREATE  VIEW   employees
       AS  SELECT  tabno, depno, name, post, born, phone
       FROM emp;

GRANT  SELECT,  INSERT,  UPDATE  ON  employees  TO  empOK;
```

Пример БД: проектная организация



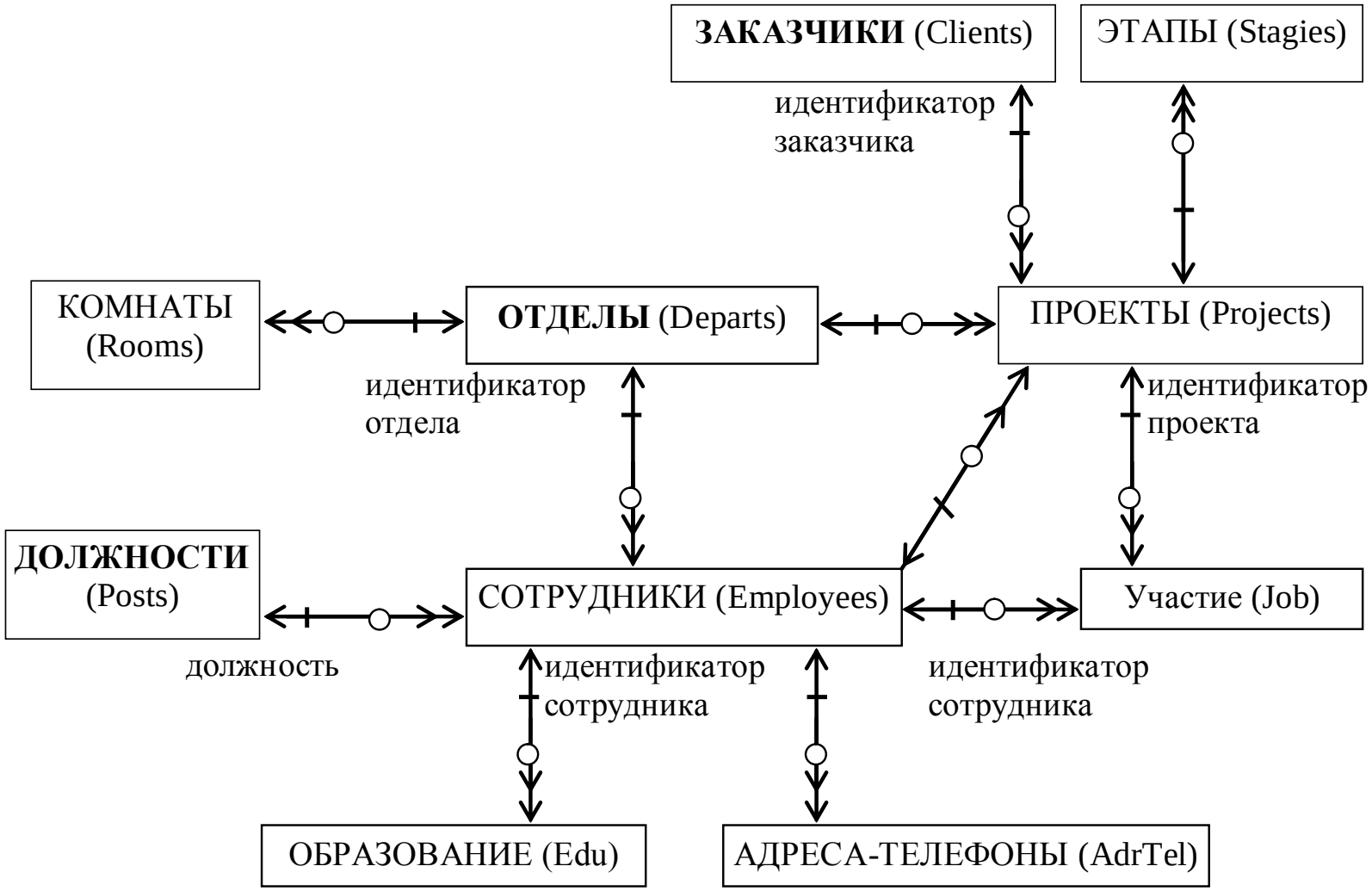
Departs – отделы,

Emp – сотрудники,

Project – проекты,

Job – участие в проектах.

Пример БД: проектная организация



Назначение прав доступа

Таблицы	Группы пользователей (роли)				
	Руководители организации	Сотрудники отд. кадров	Руководители проектов	Бухгалтеры	Участники проектов
Отделы	S	SIUD	S	S	
Комнаты	S	SUID	S	S	S
Должности	SIUD			S	
Сотрудники	S	SUID	S	S	
Адреса-телефоны	S	SUID	S	S	
Образование	S	SUID	S	S	
Заказчики	SIUD		S		
Проекты	SIUD		S		
Этапы проектов	SIUD		SUI		
Участие	S		S	S	

Назначение прав доступа через представления

1) Данные о проектах для руководителя проектов:

```
create or replace view my_projects as
select          *
from projects p
where exists (select  *  from  employees e
        where e.e_id = p.p_chief  and  e.e_login = user);
```

2) Данные об этапах проектов для руководителя проектов:

```
create or replace view my_stages as
select          s.*
from stages s
where  exists (select  *
        from  employees  e,  projects  p
        where  e.e_id = p.p_chief  and  e.e_login = user
        and  s.s_pro = p.p_abbr);
```

```
CREATE  ROLE  staff;
GRANT  ALL  ON  my_projects  TO  staff;
```

Назначение прав доступа через представления

3) Данные об участниках проектов для руководителя проектов:

```
create or replace view my_staff as
select j.*
from job j
where exists (select *
        from employees e, projects p
        where e.e_id = p.p_chief and e.e_login = user
        and j.j_pro = p.p_abbr);
```

4) Данные о других участниках проекта:

```
create or replace view my_emps as
select je.j_pro, e.e_fname||' '||e.e_lname e_name,
e_depart, e_post, e_phone, e_room
from employees e, job je
where e.e_id = je.j_emp and exists (select *
        from job jm, employees m
        where m.e_id = jm.j_emp and
        m.e_login = user and je.j_pro = jm.j_pro);
```