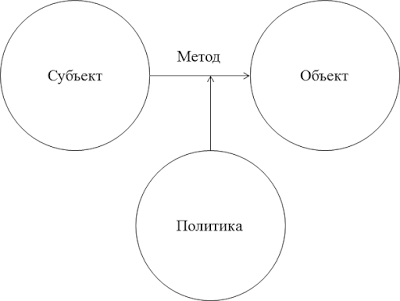
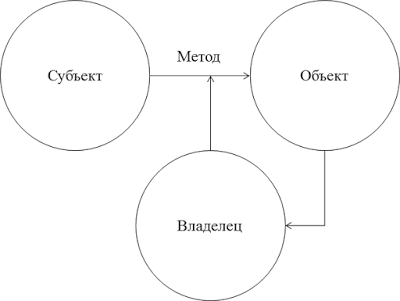
**Рекомендации по выполнению лабораторной работы**

Разграничение доступа - совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам ИС. Существуют две основные модели управления доступом:

* мандатное разграничение доступа: возможность доступа к тому или иному объекту определяется уровнем секретности объекта и уровнем доступа пользователя, которые жестко заданы для каждого пользователя и объекта. Модель обладает невысокой гибкостью и высокой трудоемкостью настройки политик безопасности, но при этом позволяет достичь высокого уровня управляемости безопасностью;
* избирательное (дискреционное) разграничение доступа: управление доступом субъектов к объектам базируется на том, что пользователи в том или ином объеме могут управлять настройками политик безопасности.

мандатное дискреционное

Простейшая одноуровневая модель безопасности данных строится на основе дискреционного принципа разграничения доступа, при котором доступ к объектам осуществляется на основе множества разрешенных отношений доступа в виде троек: «субъект доступа – тип доступа – объект доступа».

Распространенным способом формализованного представления дискреционного доступа является матрица доступа, устанавливающая перечень пользователей (субъектов) и перечень разрешенных операций (процессов) по отношению к каждому объекту базы данных (таблицы, запросы, формы, отчеты).

Принято выделять два самостоятельных подхода:

* *добровольное управление доступом* – в этом случае правами на объект управляет непосредственно субъект, являющийся его владельцем, при этом, как правило, владелец может передать права владения другому субъекту. Это позволяет достаточно гибко настраивать права, однако, такая система является децентрализованной, вследствие чего контроль соблюдения защиты данных становится весьма затруднительным;
* *принудительное управление доступом* – в этом случае управление правами на все объекты системы осуществляет один или несколько специальных доверенных субъектов, называемых администраторами. Такая система управления является централизованной, однако, менее точной, так как не позволяет субъектам-владельцам, которые имеют полную информацию об объектах, которыми они владеют, управлять правами на эти объекты.

На практике часто эти два подхода объединяются в *комбинированное управление доступом*, когда правами на объекты могут управлять как администраторы, так и их владельцы.

<https://postgrespro.ru/blog/media/229432>

В современных безопасных системах может быть реализована комбинация дискреционного (избирательного) разграничения доступа (Discretionary Access Control, DAC), ролевого разграничения доступа (Role Based Access Control, RBAC) и мандатного (принудительного, обычно многоуровневого) разграничения доступа (Mandatory Access Control, MAC). Как правило, они реализуются именно в таком порядке: следующий «поверх» предыдущего. То есть ресурс, доступный по правилам мандатного доступа, заведомо доступен по правилам дискреционного доступа, но не наоборот.

Основная система разграничения доступа в ~~Oracle~~ является система, построенная на модели избирательного управления доступом: администратор создает пользователей и назначает им привилегии на доступ к различным объектам СУБД. Привилегия – это некоторый поддерживаемый системой признак, который определяет, может ли конкретный пользователь выполнить конкретную операцию. Все привилегии в ~~Oracle~~ могут быть разделены на два класса:

* системная привилегия – право на выполнение какой-либо операции в масштабе БД;
* привилегия доступа к объекту – это разрешение пользователю на выполнение определенной операции надо определенным объектом.

В Oracle реализована поддержка принципа «безопасности по умолчанию» или принципа «наименьших привилегий»: **запрещено все, что явно не разрешено**. При этом поддерживается принцип: **для объектов, владельцем которых является пользователь, разрешены любые операции**.

Эти принципы может использовать для построения защиты:

1. Создать все объекты приложения в одной схеме и отменить пользователю-владельцу привилегию CREATE SESSION (разрешает пользователю создавать сессию – минимальная привилегия для выполнения какой-либо работы с БД);
2. Остальным пользователям и роля явно предоставить необходимые привилегии.

В этом случае при ограниченном использовании системных привилегий вроде SELECT ANY TABLE (разрешение на произвольную выборку из любой таблицы, представления или снимка любой схемы БД) и системных учетных записей гарантируется отсутствие доступа ко всем данным приложения сразу.

<https://postgrespro.ru/docs/postgresql/9.6/sql-grant>

<https://coderlessons.com/tutorials/bazy-dannykh/vyuchit-postgresql/postgresql-privilegii>

Команда GRANT имеет две основные разновидности: первая назначает права для доступа к объектам баз данных (таблицам, столбцам, представлениям, сторонним таблицам, последовательностям, базам данных, обёрткам сторонних данных, сторонним серверам, функциям, процедурным языкам, схемам или табличным пространствам), а вторая назначает одни роли членами других. Эти разновидности во многом похожи, но имеют достаточно отличий, чтобы рассматривать их отдельно.

Синтаксис запроса GRANT:

GRANT { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE | REFERENCES | TRIGGER }

[, ...] | ALL [ PRIVILEGES ] }

ON { [ TABLE ] имя\_таблицы [, ...]

| ALL TABLES IN SCHEMA имя\_схемы [, ...] }

TO указание\_роли [, ...] [ WITH GRANT OPTION ]

Значениями привилегий могут быть: SELECT, INSERT, UPDATE, DELETE, RULE, ALL.

object – имя объекта, к которому предоставляется доступ. Возможные объекты: таблица, представление, последовательность

ПУБЛИЧНЫЙ – краткая форма, представляющая всех пользователей.

ГРУППА ГРУППА – группа, которой нужно предоставить привилегии.

username – имя пользователя, которому необходимо предоставить привилегии. PUBLIC – это краткая форма, представляющая всех пользователей.

Привилегии могут быть отменены с помощью команды REVOKE.

Синтаксис запроса REVOKE

REVOKE privilege [, ...]

ON object [, ...]

FROM { PUBLIC | GROUP groupname | username }

Значениями привилегий могут быть: SELECT, INSERT, UPDATE, DELETE, RULE, ALL.

object – имя объекта, к которому предоставляется доступ. Возможные объекты: таблица, представление, последовательность

ПУБЛИЧНЫЙ – краткая форма, представляющая всех пользователей.

ГРУППА– группа, которой нужно предоставить привилегии.

username – имя пользователя, которому необходимо предоставить привилегии. PUBLIC – это краткая форма, представляющая всех пользователей.

**GRANT для объектов баз данных**

Эта разновидность команды GRANT даёт одной или нескольким ролям определённые права для доступа к объекту базы данных. Эти права добавляются к списку имеющихся, если роль уже наделена какими-то правами.

Также можно дать роли некоторое право для всех объектов одного типа в одной или нескольких схемах. Эта функциональность в настоящее время поддерживается только для таблиц, последовательностей и функций (но заметьте, что указание ALL TABLES распространяется также на представления и сторонние таблицы).

Ключевое слово PUBLIC означает, что права даются всем ролям, включая те, что могут быть созданы позже. PUBLIC можно воспринимать как неявно определённую группу, в которую входят все роли. Любая конкретная роль получит в сумме все права, данные непосредственно ей и ролям, членом которых она является, а также права, данные роли PUBLIC.

Если указано WITH GRANT OPTION, получатель права, в свою очередь, может давать его другим. Без этого указания распоряжаться своим правом он не сможет. Группе PUBLIC право передачи права дать нельзя.

Нет необходимости явно давать права для доступа к объекту его владельцу (обычно это пользователь, создавший объект), так как по умолчанию он имеет все права. (Однако владелец может лишить себя прав в целях безопасности.)

Право удалять объект или изменять его определение произвольным образом не считается назначаемым; оно неотъемлемо связано с владельцем, так что отозвать это право или дать его кому-то другому нельзя. (Однако похожий эффект можно получить, управляя членством в роли, владеющей объектом; см. ниже.) Владелец также неявно получает право распоряжения всеми правами для своего объекта.

PostgreSQL по умолчанию назначает группе PUBLIC определённые права для некоторых типов объектов. Для таблиц, столбцов, последовательностей, обёрток сторонних данных, сторонних серверов, больших объектов, схем или табличных пространств PUBLIC по умолчанию никаких прав не имеет. Для других типов объектов PUBLIC имеет следующие права по умолчанию: CONNECT и TEMPORARY (создание временных таблиц) для баз данных; EXECUTE — для функций, USAGE — для языков и типов данных (включая домены). Владелец объекта, конечно же, может отозвать (с помощью REVOKE) как явно назначенные права, так и права по умолчанию. (Для максимальной безопасности команду REVOKE нужно выполнять в транзакции, создающей объект; тогда не образуется окно, в котором другой пользователь смог бы обратиться к объекту.) Кроме того, эти изначально назначаемые права по умолчанию можно изменить, воспользовавшись командой ALTER DEFAULT PRIVILEGES.

Широко используемым, простым и эффективным способом разграничения доступа является использование представлений (view). Представление – это динамически поддерживаемая сервером выборка из одной или нескольких таблиц. В этом случае оператор SELECT, определяющий выборку, ограничивает видимые пользователем данные, а также появляется возможность ограничить данные, которые пользователь может модифицировать. Имеются следующие виды привилегий, используемых в ~~Oracle~~ исключительно для работы с представлениями:

<https://postgrespro.ru/docs/postgresql/9.6/sql-createview>

* CREATE ANY VIEW – разрешает пользователю создавать представление в любой схеме БД;
* CREATE VIEW – разрешает пользователю создавать представление в собственной схеме БД;
* DROP ANY VIEW – разрешает пользователю уничтожать любое представление в любой схеме БД.

Чтобы конструктивно работать с представлением, пользователь должен, как минимум, иметь привилегию SELECT для всех таблиц, которые участвуют в запросе, формирующем данные представления. Данная привилегия наследуется представлением для пользователя, которые его создает. Аналогичная ситуация с операторами манипулирования данными.



На рисунке представлена ситуация, в которой Пользователь 1 создает *таблицу* и *представление* (на основе таблицы). Пользователю 2 предоставляются привилегии на доступ к *представлению*, но не на доступ к *таблице*. В результате чего попытка обращения к (формально) тем же данным в базовой таблице пользователя 2 приведет к ошибке, так как привилегий недостаточно.

При рассмотрении особенностей реализации дискреционной модели доступа в ~~Oracle~~ необходимо отметить еще две возможности:

* конструкция PUBLIC используется для предоставления конкретной системной привилегии всем пользователям системы (в системах с высокими требованиями по защите информации использовать не рекомендуется)
* параметр WITH ADMIN OPTION позволяет передать некоторую системную привилегию с правом наследования.

Ролевая модель является развитием политики избирательного доступа, получившей особое развитие, начиная с 1992-го года. Она определяет особый тип политики, основанный на компромиссе между гибкостью управления доступом, характерной для дискреционных моделей и жесткостью правил контроля доступа, присущей мандатным моделям.

Ролевая модель ориентирована на описание операций с абстрактными объектами: пользователями (U), ролями (R) и привилегиями (P), а также дополнительно используется множество сеансов (S).

*Пользователями* в ролевой модели доступа могут являться люди-пользователи системы, программные сущности (процессы, приложения, службы, компоненты БД), аппаратные сущности (компьютеры и периферия, сети компьютеров).

*Роль* определяет работу над данными, которая выполняется в ИС и связана с *семантикой*, определяющей полномочия и ответственность исполнителя роли.

*Привилегия* (как уже было отмечено выше) определяет право на выполнение определенных операций с объектами системы. В ролевой модели управления доступом привилегии могут быть определены достаточно гибко – от всей базы данных, до значения элемента кортежа.

Основными элементами ролевой модели являются отношения вида «многие ко многим»:

* user assignment – пользователь может быть членом нескольких ролей, роль может содержать несколько пользователей
* permission assignment – роль может иметь много привилегий и одна привилегия может быть предоставлена нескольким ролям.

Таким образом, роль используется как посредник для передачи пользователю привилегий, что обеспечивает больший контроль технологии осуществления доступа к объектам БД.

*Сеанс* – это соответствие между конкретным пользователем и множеством ролей. Пользователь в течение сеанса активизирует некоторое подмножество множества ролей, членом которых он является. Привилегии, доступные пользователю, - это объединение привилегий всех ролей, активизированных в сеансе.

Для структурирования ролей применяются их *иерархии*, устанавливающие соответствующие зависимости между полномочиями и ответственностью должностных лиц.

В СУБД Oracle имеется средство для автоматизации работы администратора по разграничению доступа. Данное средство поддерживается как объект БД, называемый ролью (ROLE), а также предусмотрены следующие привилегии:

* CREATE ROLE – разрешает пользователю создавать роли в БД;
* GRANT ANY ROLE – разрешает пользователю предоставлять произвольную роль произвольному пользователю в БД;
* DROP ANY ROLE – разрешает пользователю уничтожать произвольную роль в БД;
* ALTER ANY ROLE – разрешает пользователю изменять произвольную роль в БД.

Управление разрешением или запрещением ролей для сессии выполняется командой SET ROLE.

Мандатная модель доступа характерна для случая, когда возможность конкретных действий с данными или документами определяется внешним, обычно глобальным собственником информации. Идея мандатного доступа состоит

**Задание на лабораторную работу**

1. Создать несколько пользователей, продемонстрировать назначение им различных привилегий (на создание объектов БД, на их просмотр и изменение).
2. Создать от имени одного из пользователей таблицы, реализовать на их базе представление и предоставить права на просмотр и изменение данных в представлении другому пользователю. Объяснить результаты.
3. Создать от имени администратора (или пользователя, обладающего достаточными привилегиями) несколько таблиц в БД.
4. Предоставить различным ролям различные привилегии (на выборку и модификацию) к созданным таблицам и продемонстрировать механизм разрешения использования ролей в рамках сеанса конкретного пользователя.