Национальный исследовательский ядерный университет «МИФИ» (Московский Инженерно–Физический Институт)
Кафедра №42 «Криптология и кибербезопасность»

Лабораторная работа №2-1: «Пользователи. Роли. Привилегии»

Цель работы

Изучение механизмов базы данных, связанных с управлением возможностями пользователей. Приобретение навыков разработки многопользовательской базы данных.

Подготовка к работе

Понятие «пользователь» в системах управления базами данных отличается от обычного понимания этого термина в операционных системах и других распространённых областях. Как правило, база данных является элементом более сложной вычислительной системы, и её пользователями являются не люди, а автоматизированные системы. Как правило, в качестве пользователя базы данных выступает серверная часть веб-сайта или службы. Типична ситуация, когда несколько экземпляров одной и той же службы или несколько разных служб осуществляют доступ к одной и той же базе данных. Эта особенность учитывается во всех основных клиент-серверных СУБД.

Поэтому представление о пользователе как об учётной записи человека или иной сущности, обладающей набором собственных прав и объектов в случае с системами управления базами данных работает плохо. Возможна (и нормальна) работа множества служб, использующих одну и ту же учётную запись и один и тот же набор объектов СУБД.

Если подойти более строго, традиционное понятие «пользователь» распадается на три аспекта: (1) группа объектов, «принадлежащих» одной и той же сущности (2) агент, наделённый правами чтения, изменения объектов в системе (3) нечто, что осуществляет подключение и работу с системой управления базами данных. На серверах СУБД эти аспекты разделены. Насколько сильно — зависит от конкретной реализации.

В PostgreSQL понятие «пользователь» отсутствует. Аспект (1) представлен отдельно, как механизм, который называется «схема» (Schema). Схема — это совокупность объектов, отнесённых к одному и тому же пространству имён. В разных схемах могут существовать два объекта с одинаковыми именами. Схемы упрощают организацию базы данных и управление разрешениями. При выполнении запросов или других операций, название схемы указывается перед названием объекта через точку. Если название объекта не указано, PostgreSQL осуществляет поиск по схемам, указанным в переменной search_path (её назначение похоже на назначение переменной окружения PATH в операционных системах.

Аспект (2) — наделение правами — осуществляется через механизм «ролей» (Roles). PostgreSQL предоставляет ряд встроенних привилегий, каждая из которых может быть предоставлена роли. Таким образом, роль — механизм для группировки привилегий. Привилегии могут быть объектными (связанными с конкретным объектом в базе данных) или системными (не привязанными к конкретному объекту, и действующими в рамках всей базы данных). Объектная привилегия — как правило, разрешение на чтение или запись конкретной таблицы, индекса и т. д.Пример системной привилегии: разрешение на подключение к серверу СУБД или разрешение на создание таблиц. Роли — не более чем контейнеры для разрешений; можно свободно присваивать роли другим ролям.

При подключении к серверу базы данных вместо имени пользователя указывается именно роль, в рамках которой подключающийся желает взаимодействовать с базой данных. Роли создаются при помощи CREATE ROLE, для изменения и удаления, как всегда, ALTER ROLE и DROP ROLE. В момент создания и изменения определяется ряд системных привилегий, которые будут предоставлены создаваемой роли. Объектные привилегии и другие роли предоставляются ролям через GRANT [what] TO [role]. В любой момент при работе можно переключаться между ролями при помощи SET ROLE, но

«перемещаться» можно только по цепочке вложенных (предоставленных одна другой) ролей. Это может быть необходимо, так как вложенные роли не всегда получают привилегии родительских ролей по умолчнанию, см. INHERIT, NOINHERIT).

Объектные привилегии могут предоставляться хозяином объекта другим ролям при помощи GRANT [privilege] ON [object] TO [role].

Чистая установка PostgreSQL предоставляет несколько стандартных ролей «из коробки». Их (а также и все созданные в дальнейшем роли) можно посмотреть в представлении системного каталога PostgreSQL: pg_catalog.pg_roles.

Аспект (3) — сущность, подключающаяся к серверу управления базами данных, — не представлена как таковая в СУБД PostgreSQL. Каждый человек и служба, который знает название роли с системной привилегией LOGIN и способен пройти авторизацию, может установить одно или несколько независимых подключений к серверу управления базами данных (сессий). Обычно, на каждую сессию выделяется рабочий процесс на сервере, который отвечает за общение с клиентом и обслуживание его запросов и команд. Несколько разных сущностей могут использовать для подключения одну и ту же роль. Одна и та же сущность может осуществлять доступ к базе данных, используя несколько ролей.

Понятие «пользователь» можно искусственно синтезировать средствами PostgreSQL, если: (1) создать роль, имя которой соответствует имя пользователя (2) создать схему, имя которой совпадает с именем пользователя (3) выдать данные для авторизации с использованием этой роли только одному человеку/службе. Такой подход используется штатно в Oracle Database, но в для PostgreSQL это нестандартно, необычно и часто не нужно.

Вопросы для самоконтроля и самостоятельного изучения

1. Чем пользователь СУБД отличается от роли?
2. Чем схема отличается от пользователя СУБД?
3. Как создать, изменить, удалить роль?
4. Каково назначение представления системного каталога pg_catalog.pg_roles?
5. Чем системные привилегии отличаются от объектных?
6. Как выдать системную привилегию пользователю?
7. Как выдать объектную привилегию пользователю?
8. Как отозвать выданную ранее привилегию у пользователя?
9. Кто имеет право выдавать объектные привилегии?
10. Кто имеет право выдавать системные привилегии?
11. Если дать роли Б роль А, наследует ли роль Б права роли А?
12. Если дать роли Б роль А, а потом отозвать привилегию у роли А, скажется ли это на роли Б?
13. Что означает ключевое слово LOGIN?
14. Что означает ключевое слово INHERIT?
15. Что означает ключевое слово CREATEDB?
16. Что означает ключевое слово CREATEROLE?
17. Что означает ключевое слово SUPERUSER?
18. Что означает ключевое слово WITH ADMIN?
19. Что означает ключевое слово (не SQL-инструкция) SET?
20. Что означает ключевое слово LOGIN?
21. Приведите конкретный пример использования SET ROLE;
22. Есть ли пользователи в чистой установке PostgreSQL?
23. Сколько подключений к серверу БД (сессий) может быть установлено одной и той же ролью?
24. Кто такой «хозяин объекта» (owner). Как его определить?
25. Как сменить хозяина объекта?
26. Какими дополнительными привилегиями обладает хозяин объекта?

Ход работы

1. Определить, в какой схеме находятся таблицы Вашей базы данных. Следует ли изменить схему? Следует ли создать несколько отдельных схем для выбранной предметной области? Почему?
2. Определить, какие роли нужны для нормального функционирования Вашей базы данных. Какие системные и объектные привилегии потребуются каждой роли? Понадобятся ли вложенные роли?
3. Создать роли и выдать им необходимые объектные и системные привилегии;
4. Проверить по представлению системного каталога pg_catalog.pg_roles, что все нужные роли были созданы и обладают корректным набором привилегий;
5. Попробовать подключиться от лица каждой роли (из тех, которым разрешено подключение к серверу БД). Убедиться, что роль имеет доступ к разрешённым данным и не имеет доступа ко всем остальным.
6. Оформить отчёт.

Оформление отчёта

1. Титульный лист: название института, название лабораторной работы, имя, фамилия, номер группы, год,…
2. Лист с диаграммой отношения сущностей (полная страница);
3. Ответы на вопросы, поставленные по Ходу работы;
4. Заключение: краткое описание проделанной работы;
5. Приложение: SQL-инструкции для создания, проверки и настройки ролей;
6. Приложение: SQL-инструкции, использованные при проверке привилегий и прав доступа.