**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Направление подготовки: Выберите элемент.**

**Образовательная программа: Выберите элемент.**

**Дисциплина:**

«Информационная безопасность баз данных»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

«Разграничение доступа»

Содержание

[Введение 3](#_Toc194578375)

[1 Разграничение доступа 4](#_Toc194578376)

[1.1 Ход работы 4](#_Toc194578377)

[Заключение 11](#_Toc194578378)

[Список использованных источников 12](#_Toc194578379)

Введение

Цель работы – для достижения поставленной цели необходимо решить следующие задачи:

* создать таблицы и пользователей для работы;
* раздать различные права пользователям;
* создать схемы, представления и триггеры;
* сделать выводы о проделанной работе.

# Разграничение доступа

## Ход работы

1. Для выполнения последующих заданий создадим две таблицы и четыре пользователя.

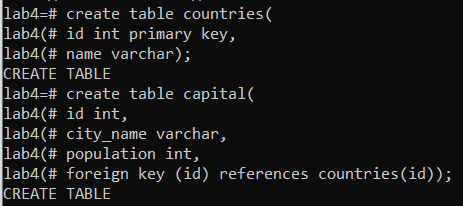


Рисунок 1 – создание таблиц

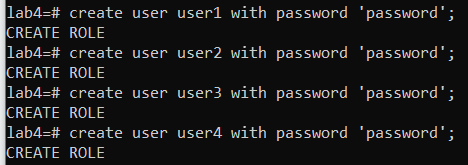


Рисунок 2 – создание пользователей

1. Выдадим права 3 пользователям. Пользователю User1 дадим полный доступ к таблице. User2 дадим право на вставку, select-запросы и обновление значений в таблицах. User3 получит право на удаление строк из таблиц, а также возможность делегировать свои права любому пользователю.



Рисунок 3 – выдача прав первому пользователю



Рисунок 4 – выдача прав второму пользователю



Рисунок 5 – выдача прав третьему пользователю

Просмотрим настройку прав, установленные для таблиц на текущий момент, с помощью команды \dp.

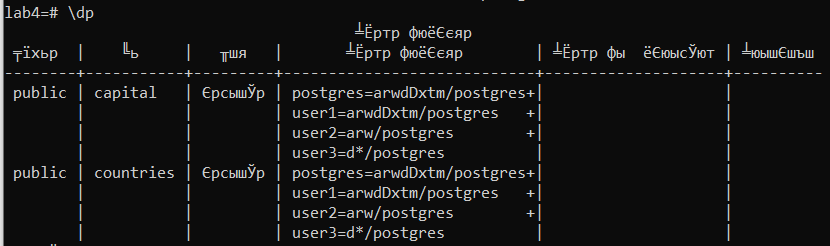


Рисунок 6 – права, установленные для таблиц

1. Предоставим право на удаление от пользователя User3 пользователю User4, для этого подключимся к базе данных под пользователем User3, и после этого уже будет возможность передачи прав. После выполнения нужных заданий вернемся обратно в учетную запись супер-пользователя.



Рисунок 7 – вход как пользователь User3



Рисунок 8 – передача прав



Рисунок 9 – возвращение к роли супер-пользователя

Проверим все выданные права.

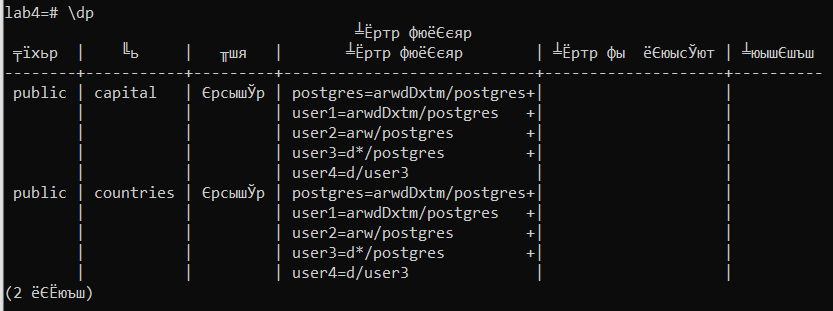


Рисунок 10 – проверка выданных прав

1. Отменим все предоставленные выше права с помощью команды revoke.

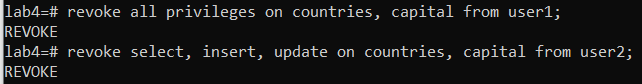


Рисунок 11 – отмена прав для пользователя 1 и 2

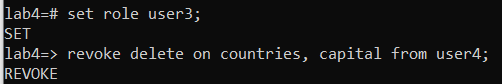


Рисунок 12 – отмена прав для пользователя 4

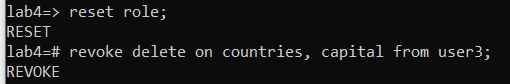


Рисунок 13 – отмена прав для пользователя 3

Проверим, что все предоставленные права были отменены.

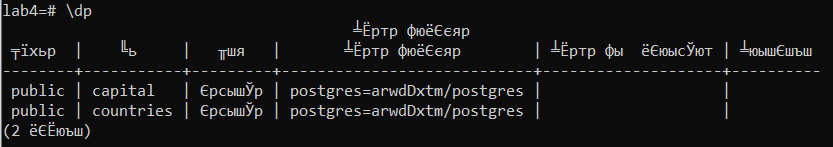


Рисунок 14 – проверка отмены всех прав

1. Создадим подсхему авторизации для User1 и User2 с различным набором таблиц (с помощью служебного слова AUTHORIZATION у команды CREATE SCHEMA).

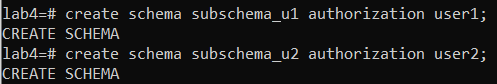


Рисунок 15 – создание схем

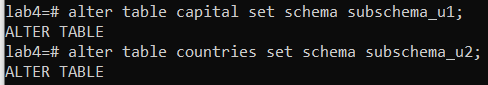


Рисунок 16 – добавление таблиц в схемы

Проверим созданные нами схемы с помощью команды \dn, которая выводит все существующие схемы.

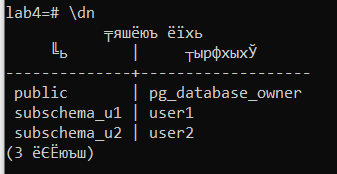


Рисунок 17 – проверка созданных схем

1. Создадим представление как объединенный набор столбцов из разных таблиц. А именно столбец с названием страны из таблицы со странами со столбцами с названием столицы и численностью населения из таблицы со столицами.

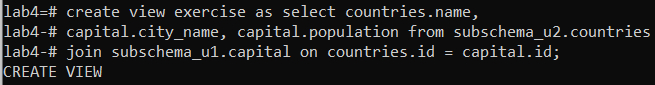


Рисунок 18 – создание представления

Проверим, что таблица создалась, выведя ее на экран.

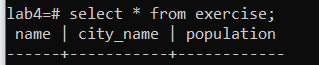


Рисунок 19 – вывод таблицы

С помощью команды grant ограничим доступ к представлению. Пусть пользователь user1 получит право на select-запросы, user2 получит все привилегии, а user3 – право на обнавление.

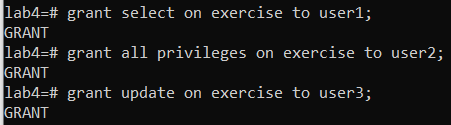


Рисунок 20 – ограничение доступа к представлению

Проверим выданные права.

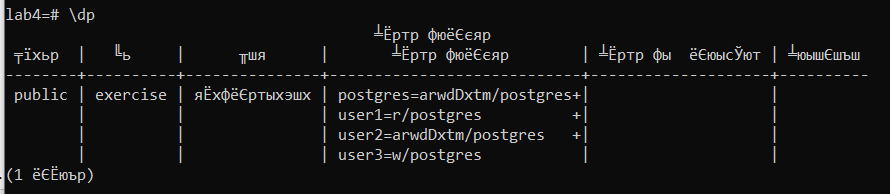


Рисунок 21 – проверка выданных прав

1. Настроим безопасность на уровне строк (RLS) для таблицы со столицами, политика создана на основе текущего пользователя, то есть пользователь postgres.





Рисунок 22 – настройка безопасности на уровне строк

Проверим, что безопасность действительно работает, попробуем обратиться к таблице от user1, это у нас не получится, возникает ошибка, так как у этого пользователя нет доступа. Но если мы попробуем обратиться к таблице от имени пользователя postgres, то все получится.

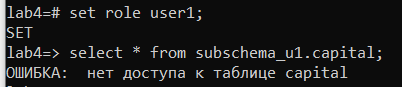


Рисунок 23 – обращение к таблице пользователем user1

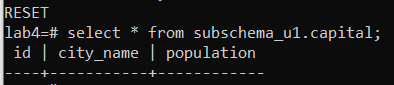


Рисунок 24 – обращение к таблице пользователем postgres

1. Создадим триггер для регистрации вставки, обновления и удаления содержимого в определенных таблицах. Для того сначала создадим таблицу с логам, куда и будут записываться все изменения.

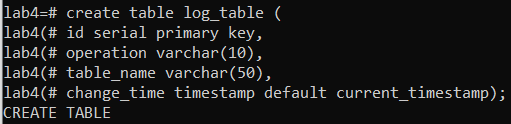


Рисунок 25 – создание таблицы с логами

Напишем триггерную функцию, а затем сами триггеры для двух исходных таблиц.

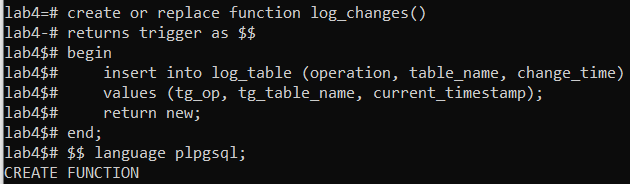


Рисунок 26 – триггерная функция

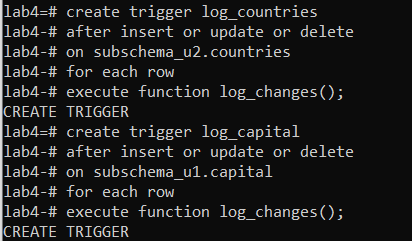


Рисунок 27 – создание триггеров

Проверим работу триггера, добавим новые записи в таблицу, обновим и удалим некоторые записи.

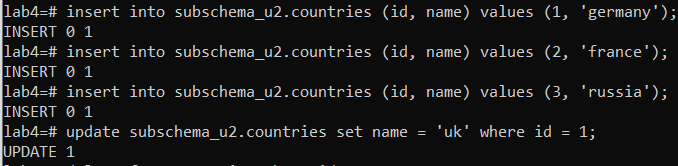




Рисунок 28 – изменения записей таблиц

Выведем таблицу с логами и увидим, что действительно все изменения зафиксированы в ней.

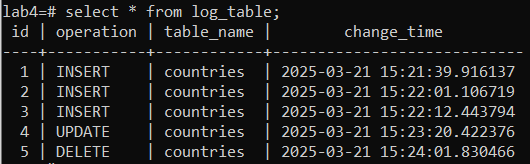


Рисунок 29 – таблица с логами

Заключение

В ходе данной лабораторной работы было изучено разграничение доступа в БД на языке SQL. Были выданы разные права пользователям, написаны процедуры и триггер для заполнения таблицы с логами. В ходе выполнения лабораторной работы были выполнены все задачи и достигнуты поставленные цели.

Список использованных источников

1. Основы технологий баз данных: учебное пособие Б. А. Новиков, Е. А. Горшкова, Н. Г. Графеева; под ред. Е. В. Рогова. — 2-е изд. — М.: ДМК Пресс, 2020. — 582 с.