МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение

высшего образования

**«Северный (Арктический) федеральный университет имени М.В. Ломоносова»**

|  |
| --- |
| Высшая школа информационных технологий и автоматизированных систем |

**ЛАБОРАТОРНАЯ РАБОТА №11-12**

|  |  |  |
| --- | --- | --- |
| По дисциплине: | | Защита информации в системах управления базами данных |
|  | | |
|  | | |
| На тему: | Защита хоста | |
|  | | |
|  | | |

|  |  |
| --- | --- |
|  | Выполнил обучающийся:  Грозов Илья Владимирович |
|  |  |
|  | Направление подготовки / специальность:  10.03.01 Информационная безопасность |
|  |  |
|  | Курс: 3 |
|  | Группа: 151113 |
|  | Руководитель: Зубарев Александр Андреевич, ст. |
|  | преподаватель |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Отметка о зачете |  |  |  |  |
|  |  |  |  |  |
| Руководитель |  |  |  | А.А. Зубарев. |
|  |  |  |  |  |

Архангельск 2024

##### Задание

Получить практический навык защиты сети при эксплуатации СУБД

ХОД РАБОТЫ

**1 УСТАНОВКА POSTGRES**

Установим postgres используя команду: sudo apt-get install postgresql postgres-contrib. Установка postgres отображена на рисунке 1

Изображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описание

Рисунок 1 – Установка postgres

При помощи команды: sudo systemctl status postgresql проверим запущен ли системный процесс. Проверка запуска отображена на рисунке 2

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 2 – Проверка запуска

Войдем в оболочку postgres при помощи команды: sudo -u postgres psql. Вход в оболочку postgres отображен на рисунке 3

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – Вход в оболочку postgres

**2 РАЗВЕРТЫВАНИЕ БАЗЫ ДАННЫХ**

Загрузим архив с предложенной базой данных при помощи команды: wget <https://edu.postgrespro.ru/demo_medium.zip>. Загрузка архива с базой данных отображена на рисунке 4

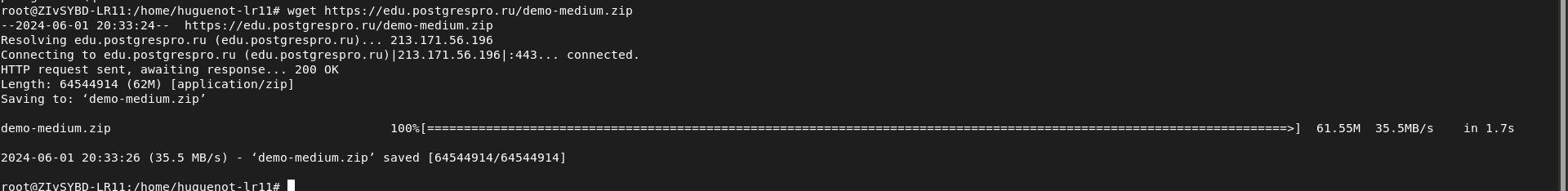
****

Рисунок 4 – Загрузка архива с базой данных

Распакуем архив с базой данных при помощи команды: unzip demo-medium.zip. Распаковка архива отображена на рисунке 5

**Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание**

Рисунок 5 – Распаковка архива

Переместим распакованную базу данных в более удобное место при помощи команды mv. Перемещение базы данных отображено на рисунке 6

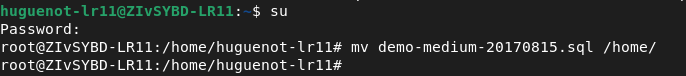
****

Рисунок 6 – Перемещение базы данных

При помощи команды sudo -u postgres psql выполним вход в оболочку и запустим скрипт для базы данных при помощи команды: \i /home/demo-medium-20170815.sql. Запуск скрипта для базы данных отображен на рисунке 7

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 7 – Запуск скрипта для базы данных

База данных было автоматически восстановлена из sql скрипта. Восстановленная база данных отображена на рисунке 8

**Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание**

Рисунок 8 – Восстановленная база данных из sql скрипта

**3 РАЗГРАНИЧЕНИЕ ПРАВ ПОЛЬЗОВАТЕЛЕЙ**

3.1 Действия перед разграничением

Перед созданием новых пользователей и разграничением их прав, выполним просмотр списка всех существующих таблиц. Пользователи с правми будут созданы на их основе. Просмотр списка существующих таблиц выполняется при помощи команды \dt в оболочке postgres базы данных demo. Существующие таблицы отображены на рисунке 9

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

Рисунок 9 – Существующие таблицы

В развернутой базе данных demo существует 8 таблиц:

* aircrafts\_data;
* airports\_data;
* boarding\_passes;
* bookings;
* flights;
* seats;
* ticket\_flights;
* tickets

Для существующих таблиц возможно минимально создать 4 пользователя:

* администратор – администратор базы данных, имеет доступ ко всем таблицам;
* диспетчер – имеет доступ к таблицам, необходимым для организации воздушного пространства;
* контроллер – имеет доступ к таблицам, необходимым для произведения посадки на борт;
* продавец – имеет доступ к таблицам, необходимым для получения информации о самолетах, маршрутах, бронировании и продажи билетов

На основе приеденных данных составим таблицу для разграничения прав пользователей

Таблица 1 – Разграничение прав пользователей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблицы БД | | Пользователи БД | | | |
| administrator | dispatcher | controller | salesman |
| 1 | aircrafts\_data |  |  |  |  |
| 2 | airports\_data |  |  |  |  |
| 3 | boarding\_passes |  |  |  |  |
| 4 | bookings |  |  |  |  |
| 5 | flights |  |  |  |  |
| 6 | seats |  |  |  |  |
| 7 | ticket\_flights |  |  |  |  |
| 8 | tickets |  |  |  |  |

3.2 Создание пользователей и разграничение прав

Создадим пользователя administrator используя составленную таблицу 1. Выдадим пользователю administrator необходимые права для доступа к таблицам базы данных при помощи команд, отраженных в листинге 1

Листинг 1 – Команды для создания пользователя administrative и выдача прав

CREATE USER administrator WITH PASSWORD 'administrator`;

GRANT INSERT, UPDATE, DELETE ON aircrafts\_data TO administrator;

GRANT INSERT, UPDATE, DELETE ON airports\_data TO administrator;

GRANT INSERT, UPDATE, DELETE ON boarding\_passes TO administrator;

GRANT INSERT, UPDATE, DELETE ON bookings TO administrator;

GRANT INSERT, UPDATE, DELETE ON flights TO administrator;

GRANT INSERT, UPDATE, DELETE ON seats TO administrator;

GRANT INSERT, UPDATE, DELETE ON ticket\_flights TO administrator;

GRANT INSERT, UPDATE, DELETE ON tickets TO administrator;

Создание пользователя administrator и выдача прав к таблицам отображена на рисунке 10

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 10 – Создание пользователя administrator и выдача прав к таблицам

Создадим пользователя dispatcher используя составленную таблицу 1. Выдадим пользователю dispatcher необходимые права для доступа к таблицам базы данных при помощи команд, отраженных в листинге 2

Листинг 2 – Команды для создания пользователя dispatcher и выдача прав

CREATE USER dispatcher WITH PASSWORD 'dispathcer';

GRANT SELECT, INSERT, UPDATE, DELETE ON aircrafts\_data TO dispatcher;

GRANT SELECT, INSERT, UPDATE, DELETE ON airports\_data TO dispatcher;

GRANT SELECT, INSERT, UPDATE, DELETE ON boarding\_passes TO dispatcher;

Создание пользователя dispatcher и выдача прав к таблицам отображена на рисунке 11

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 11 – Создание пользователя dispatcher и выдача прав к таблицам

Создадим пользователя controller используя составленную таблицу 1. Выдадим пользователю controller необходимые права для доступа к таблицам базы данных при помощи команд, отраженных в листинге 3

Листинг 3 – Команды для создания пользователя controller и выдача прав

CREATE USER controller WITH PASSWORD 'controller';

GRANT SELECT, INSERT, UPDATE, DELETE ON boards\_passes TO controller;

GRANT SELECT, INSERT, UPDATE, DELETE ON flights TO controller;

GRANT SELECT, INSERT, UPDATE, DELETE ON seats TO controller;

GRANT SELECT, INSERT, UPDATE, DELETE ON ticket\_flights TO controller;

GRANT SELECT ON aircrafts\_data TO controller;

GRANT SELECT ON airports\_data TO controller;

Создание пользователя controller и выдача прав к таблицам отображена на рисунке 12

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 12 – Создание пользователя controller и выдача прав к таблицам

Создадим пользователя salesman используя составленную таблицу 1. Выдадим пользователю salesman необходимые права для доступа к таблицам базы данных при помощи команд, отраженных в листинге 4

Листинг 4 – Команды для создания пользователя salesman и выдача прав

CREATE USER salesman WITH PASSWORD ' salesman';

GRANT SELECT, INSERT, UPDATE, DELETE ON boards\_passes TO salesman;

GRANT SELECT, INSERT, UPDATE, DELETE ON bookings TO salesman;

GRANT SELECT, INSERT, UPDATE, DELETE ON flights TO salesman;

GRANT SELECT, INSERT, UPDATE, DELETE ON seats TO salesman;

GRANT SELECT, INSERT, UPDATE, DELETE ON ticket\_flights TO salesman;

GRANT SELECT, INSERT, UPDATE, DELETE ON tickets TO salesman;

Создание пользователя salesman и выдача прав к таблицам отображена на рисунке 13

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 13 – Создание пользователя salesman и выдача прав к таблицам

Нам требуется также выдать права на доступ к схемам созданным пользователям при помощи команд отраженных в листинге 5

Листинг 5 – Команды для выдачи доступа пользователям к схемам

GRANT USAGE ON SCHEMA bookings TO salesman;

GRANT USAGE ON SCHEMA bookings TO controller;

GRANT USAGE ON SCHEMA bookings TO dispatcher;

GRANT USAGE ON SCHEMA bookings TO administrator;

Выдача доступа к схемам отображена на рисунке 14

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 14 – Выдача доступа к схемам

3.3 Проверка разграничения прав пользователей

Для возможности авторизации под другими пользователями нам необходимо сконфигурировать файл pg\_hba.conf открыв его при помощи команды: sudo nano /etc/postgresql/13/main/pg\_hba.conf. Открытие файла конфигуарции отображено на рисунке 15



Рисунок 15 – Открытие файла конфигурации

Изменим все существующие значения peer на md5 кроме пользователя postgres, на приведенном рисунке это значение было изменено ошибочно. Настройки авторизации для пользователей отображены на рисунке 16

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 16 – Настройки авторизации для пользователей

Выполним перезагрузку сервиса postgres при помощи команды sudo service postgresql restart. Выполнение перезагрузки сервиса postgres отображено на рисунке 17



Рисунок 17 – Выполнение перезагрузки сервиса postgres

3.3.1 Проверка пользователя salesman

Для проверки настроенных прав пользователя salesman выполним авторизацию под ним и отобразим список таблиц при помощи команды \dt. Авторизация под пользователем salesman и отображение списка таблиц отображено на рисунке 18

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 18 – Авторизация под пользователем salesman и отображение списка таблиц

Из таблицы 1 мы знаем, что у нас есть доступ к таблице tickets. Выполним запрос в данной таблице для извлечения первых 10 значений при помощи команды: SELECT \* FROM tickets LIMIT 10; Выполнение запроса к таблице tickets пользователя salesman отображено на рисунке 19

Изображение выглядит как текст, снимок экрана, черный, Шрифт

Автоматически созданное описание

Рисунок 19 – Выполнение запроса к таблице tickets пользователя salesman

Доступ есть, данные были извлечены. Из таблицы 1 мы знаем, что для пользователя salesman отсутствует доступ к таблице airports\_data. Выполним запрос в данной таблице для извлечения первых 10 значений при помощи команды: SELECT \* FROM airports\_data LIMIT 10; чтобы убедиться в том, что у данного пользователя нет прав на просмотр. Выполнение запроса к таблице airports\_data пользователя salesman отображено на рисунке 20

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание

Рисунок 20 – Выполнение запроса к таблице airports\_data пользователя salesman

3.3.2 Проверка пользователя controller

Для проверки настроенных прав пользователя controller выполним авторизацию под ним и отобразим список таблиц при помощи команды \dt. Авторизация под пользователем controller и отображение списка таблиц отображено на рисунке 21

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 21 – Авторизация под пользователем controller и отображение списка таблиц

Из таблицы 1 мы знаем, что у нас есть доступ к таблице seats. Выполним запрос в данной таблице для извлечения первых 10 значений при помощи команды: SELECT \* FROM seats LIMIT 10; Выполнение запроса к таблице seats пользователя controller отображено на рисунке 22

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 22 – Выполнение запроса к таблице seats пользователя controller

Доступ есть, данные были извлечены. Из таблицы 1 мы знаем, что для пользователя controller отсутствует доступ к таблице bookings. Выполним запрос в данной таблице для извлечения первых 10 значений при помощи команды: SELECT \* FROM bookings LIMIT 10; чтобы убедиться в том, что у данного пользователя нет прав на просмотр. Выполнение запроса к таблице bookings пользователя controller отображено на рисунке 23

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

Рисунок 23 – Выполнение запроса к таблице bookings пользователя controller

3.3.3 Проверка пользователя dispatcher

Для проверки настроенных прав пользователя dispatcher выполним авторизацию под ним и отобразим список таблиц при помощи команды \dt. Авторизация под пользователем dispatcher и отображение списка таблиц отображено на рисунке 24

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 24 – Авторизация под пользователем dispatcher и отображение списка таблиц

Из таблицы 1 мы знаем, что у нас есть доступ к таблице airports\_data. Выполним запрос в данной таблице для извлечения первых 10 значений при помощи команды: SELECT \* FROM airports\_data LIMIT 10; Выполнение запроса к таблице airports\_data пользователя dispatcher отображено на рисунке 25

Изображение выглядит как текст, снимок экрана, черный, Шрифт

Автоматически созданное описание

Рисунок 25 – Выполнение запроса к таблице airports\_data пользователя dispatcher

Доступ есть, данные были извлечены. Из таблицы 1 мы знаем, что для пользователя dispatcher отсутствует доступ к таблице tickets\_flights. Выполним запрос в данной таблице для извлечения первых 10 значений при помощи команды: SELECT \* FROM tickets\_flights LIMIT 10; чтобы убедиться в том, что у данного пользователя нет прав на просмотр. Выполнение запроса к таблице tickets\_flights пользователя dispatcher отображено на рисунке 26

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание

Рисунок 26 – Выполнение запроса к таблице tickets\_flights пользователя dispatcher

3.3.3 Проверка пользователя administrator

Для проверки настроенных прав пользователя administrator выполним авторизацию под ним и отобразим список таблиц при помощи команды \dt. Авторизация под пользователем administrator и отображение списка таблиц отображено на рисунке 27

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 27 – Авторизация под пользователем administrator и отображение списка таблиц

Из таблицы 1 мы знаем, что у нас есть доступ ко всем таблицам. Разграничение прав доступа не выполнялось, но функция SELECT была упущена и не добавлена пользователю administrator

**4 НАСТРОЙКА РЕЗЕРВНОГО КОПИРОВАНИЯ**

Для настройки резервного копирования создадим папку backup\_demo при помощи команды: mkdir backup\_demo. Создание папки для резервного копирования отображено на рисунке 28

****

Рисунок 28 – Создание папки для резервного копирования

При помощи команды: touch backup\_demo\_lr.sh создадим файл будущего скрипта для выполнения резервного копирования и откроем его при помощи команды sudo nano backup\_demo\_lr.sh. Создание файла резервного копирования и открытие скрипта для редактирования отображено на рисунке 29

**Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание**

Рисунок 29 – Создание файла резервного копирования и открытие скрипта для редактирования

Напишем скрипт для резервного копирования. Написанный скрипт для резервного копирования отображен на рисунке 30

**Изображение выглядит как текст, Шрифт, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание**

Рисунок 30 - Написанный скрипт для резервного копирования

Проверим написанный скрипт для резервного копирования запустив его при помощи команды: ./backup\_demo\_lr.sh. Запуск скрипта резервного копирования отображен на рисунке 31

****

Рисунок 31 – Запуск скрипта резервного копирования

Во избежание ошибки отображенной на рисунке 32 необходимо изменить метод аутентификации для пользователя. Ошибка аутентификации пользователя отображена на рисунке 32

****

Рисунок 32 – Ошибка аутентификации пользователя

После изменения метода аутентификации пользователя в указанной папке будет располагаться файл базы данных после резервного копирования. Файл базы данных после резервного копирования отображен на рисунке 33

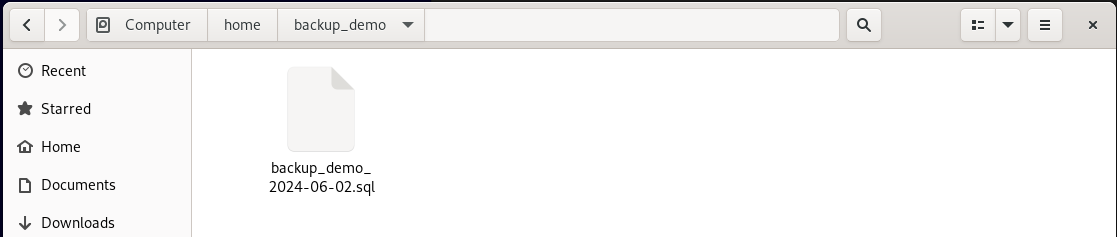
****

Рисунок 33 - Файл базы данных после резервного копирования

Добавим скрипт резервного копирования базы данных для автоматического выполнения в crontab. Откроем crontab при помощи команды: crontab -e. Открытие crontab отображено на рисунке 34

****

Рисунок 34 – Открытие crontab

Отредактируем crontab внеся строку следующего содержания: 0 0 \* \* \* /home/backup\_demo\_lr.sh. Данные изменения позволят автоматически выполнять резервное копирование базы данных каждый день в 00:00. Редактирование crontab отображено на рисунке 35

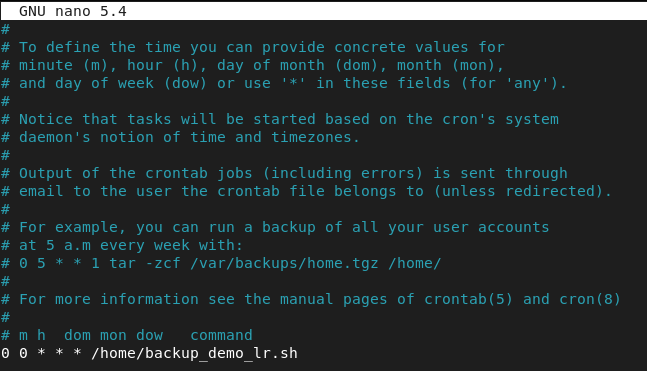
****

Рисунок 35 – Редактирование crontab

**5 ХЭШИРОВАНИЕ ХРАНЯЩИХСЯ ДАННЫХ ПО SHA-1**

Выполним хэширование хранящихся данных по sha-1. Изменим выбранную таблицу airports\_data добавив в нее столбец sha1\_hashed при помощи команды: ALTER TABLE airports\_data ADD COLUMN sha1\_hashed TEXT; Изменение таблицы airports\_data отображено на рисунке 36

****

Рисунок 36 – Изменение таблицы airports\_data

Для выполнения хэширование хранящихся данных по sha-1 нам необходимо добавить расширение pgcrypto при помощи команды: CREATE EXTENSION pgcrypto; Добавление расширения pgcrypto отображено на рисунке 37

****

Рисунок 37 – Добавление расширения pgcrypto

Обновим таблицу airports\_data при помощи команды: UPDATE airports\_data SET sha1\_hashed = encode(digest(airport\_name::text, 'sha1'), 'hex'); Обновление таблицы airports\_data отображено на рисунке 38

****

Рисунок 38 – Обновление таблицы airports\_data

При помощи команды: SELECT \* FROM airports\_data LIMIT 10; проверим, что хэширования данных было выполнено. Выполненное хэширования отображено на рисунке 39

**Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание**

Рисунок 39 – Выполненное хэширования

**6 СОСТАВЛЕНИЕ ЗАПРОСОВ ДОСТУПА**

Составим простые команды для выполнения запросов доступа.

При помощи команды: SELECT \* FROM aircrafts\_data LIMIT 10; получим первые 10 строк таблицы aircrafts\_data. Получение первых 10 строк таблицы aircrafts\_data отображено на рисунке 40

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Рисунок 40 – Получение первых 10 строк таблицы aircrafts\_data

При помощи команды: SELECT \* FROM airports\_data LIMIT 10; получим первые 10 строк таблицы airports\_data. Получение первых 10 строк таблицы airports\_data отображено на рисунке 41

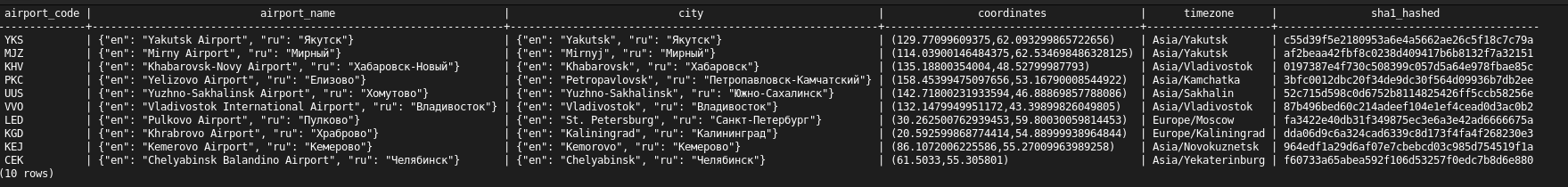


Рисунок 41 – Получение первых 10 строк таблицы airports\_data

При помощи команды: SELECT \* FROM boarding\_passes LIMIT 10; получим первые 10 строк таблицы boarding\_passes. Получение первых 10 строк таблицы boarding\_passes отображено на рисунке 42

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 42 – Получение первых 10 строк таблицы boarding\_passes

При помощи команды: SELECT \* FROM bookings LIMIT 10; получим первые 10 строк таблицы bookings. Получение первых 10 строк таблицы bookings отображено на рисунке 43

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 43 – Получение первых 10 строк таблицы bookings

При помощи команды: SELECT \* FROM flights LIMIT 10; получим первые 10 строк таблицы flights. Получение первых 10 строк таблицы flights отображено на рисунке 43

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Рисунок 43 – Получение первых 10 строк таблицы flights

При помощи команды: SELECT \* FROM seats LIMIT 10; получим первые 10 строк таблицы seats. Получение первых 10 строк таблицы seats отображено на рисунке 44

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 44 – Получение первых 10 строк таблицы seats

При помощи команды: SELECT \* FROM ticket\_flights LIMIT 10; получим первые 10 строк таблицы ticket\_flights. Получение первых 10 строк таблицы ticket\_flights отображено на рисунке 45

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание

Рисунок 45 – Получение первых 10 строк таблицы ticket\_flights

При помощи команды: SELECT \* FROM tickets LIMIT 10; получим первые 10 строк таблицы tickets. Получение первых 10 строк таблицы tickets отображено на рисунке 46

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Рисунок 46 – Получение первых 10 строк таблицы tickets

КОНТРОЛЬНЫЕ ВОПРОСЫ

**1. Что такое хэш функция?**

Хеш — это уникальная строка символов, полученная после применения специальной хеш-функции к определенным входным данным

**2. Какие алгоритмы поддерживает pgsql pro?**

Шифрование данных:

* AES;
* RSA;

Хэширование данных:

* MD5;
* SHA-1;
* SHA-256

**3. Какая модель разграничения прав доступа реализована в pgsql pro в чем ее преимущества?**

PGSQL Pro реализует модель разграничения прав доступа на основе ролей (Role-Based Access Control, RBAC). Преимущества данной модели: управление доступом на основе ролей использует принцип наименьших привилегий, поэтому пользователи располагают только теми правами, которые им нужны при исполнении своей работы, автоматические настройки и создание групп облегчают назначение прав, охватывают всю систему, снижают риски неправильно назначенной роли

ВЫВОД

В ходе выполнения лабораторной работы по теме: «Защита хоста» получили практический навык защиты сети при эксплуатации СУБД