|  |
| --- |
|  |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

**Институт кибербезопасности и цифровых технологий**

Дисциплина

«Специальные информационные технологии

в правоохранительной деятельности»

**Отчет**

по практической работе №2-6 на тему:

**«Работа со сторонними базами данными.**

**Построение и оптимизация»**

Выполнил:

Студент 3 курса

Группа ББСО-02-20

Шифр 20Б0910

Шамрай Максим Александрович

Шифр 20Б1686

Левина Анна Александровна

Москва, 2023

**Задание**

1. Разработка ER диаграммы и реализация структуры базы данных используя PostgreSQL. База данных должна быть нормализована и лишена аномалий удаления, добавления и редактирования.
2. Заполнение тестовыми данными.
3. Создание ролей и групп ролей и назначение привилегий.
4. Настройка политик безопасности.
5. Автоматизация функционала за счёт добавление функций, процедур, триггеров.
6. Генерация отчетов.

**Задание 1 - 2**

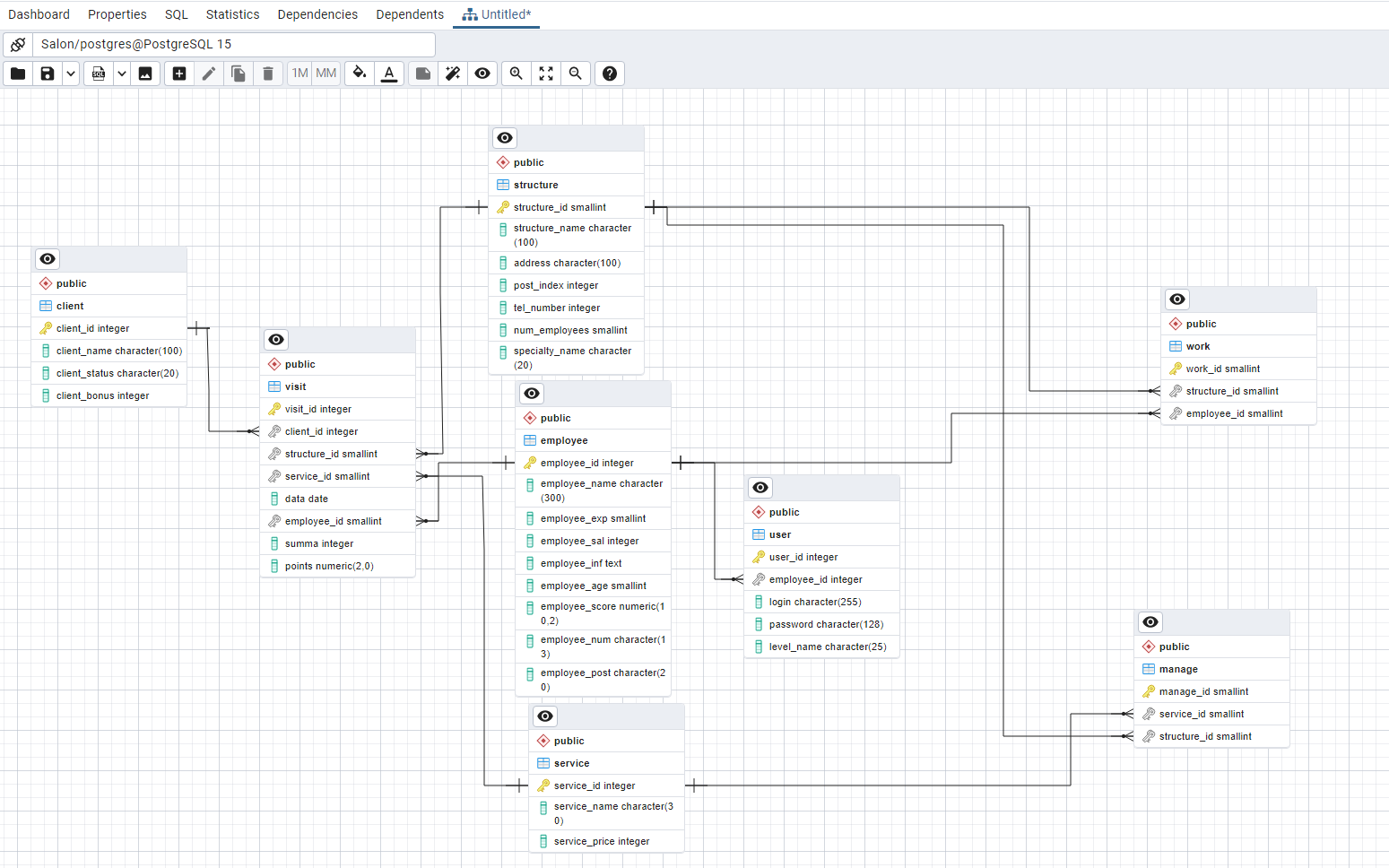
****

Рисунок 1 ER диаграмма

Рассмотрим подробнее каждую таблицу

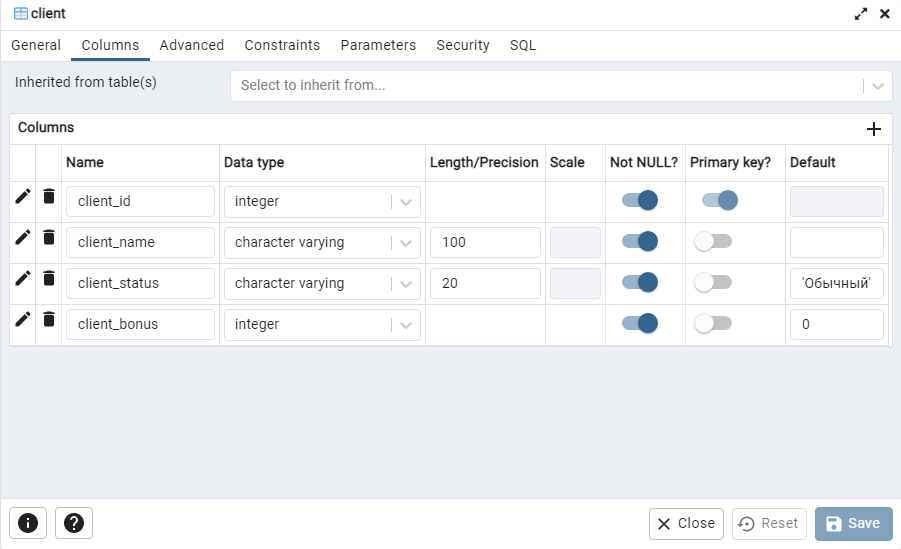
*  Таблица «Client»

Рисунок 2 Структура таблицы Client

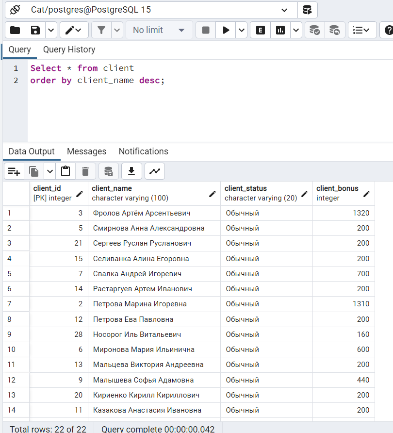
Данная таблица хранит информацию о клиенте. При добавлении нового клиента значения статуса и кол-во бонусов заполняются автоматически.

Рисунок 3 Наполнение таблицы Client

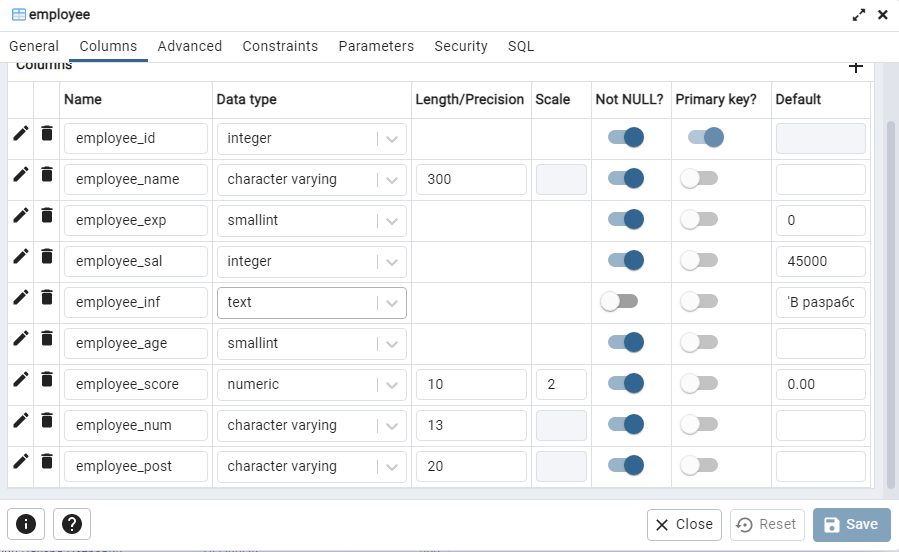
* Таблица «Employee»

Рисунок 4 Структура таблицы Employee

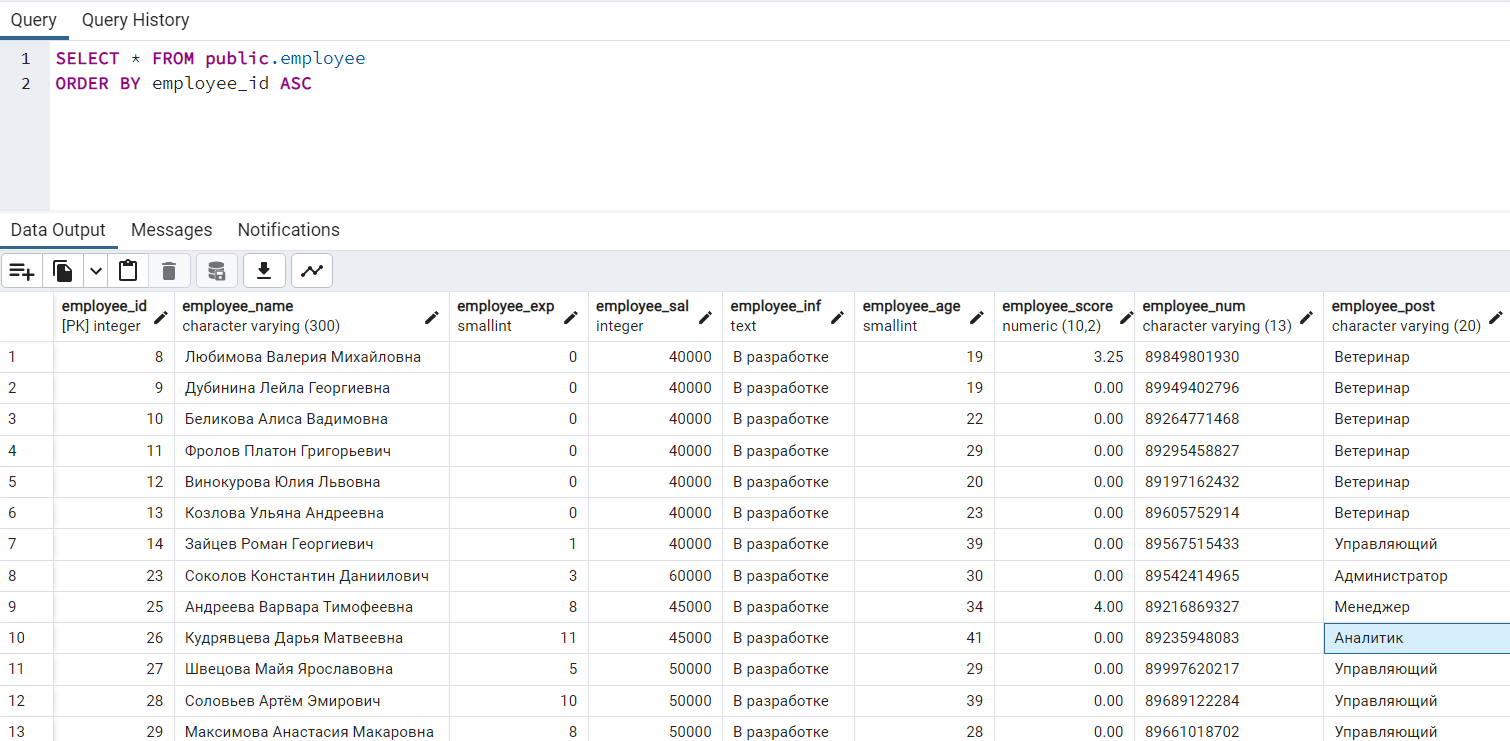
Данная таблица хранит информацию о сотрудниках. При добавлении нового сотрудника значения заработной платы, личной информации и рейтинге заполняются автоматически.

Рисунок 5 Наполнение таблицы Employee

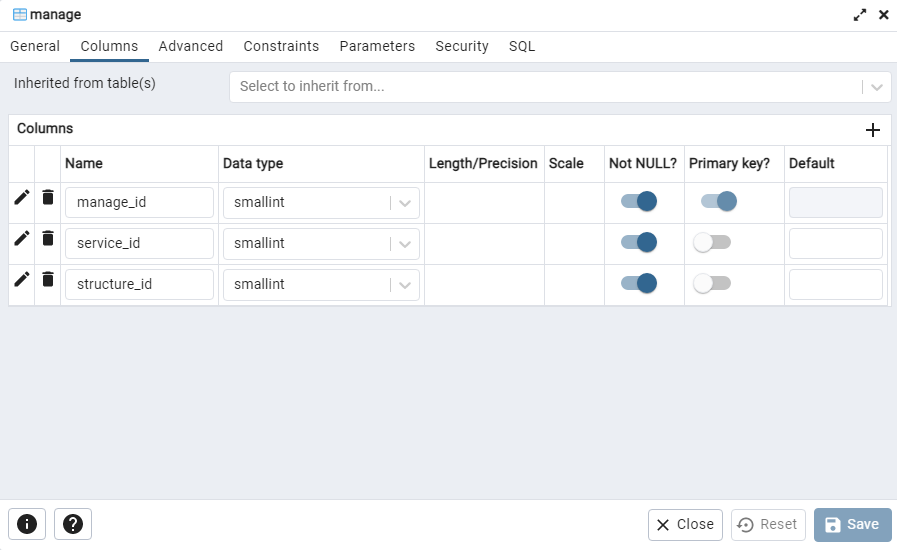
* Таблица «Manage»

Рисунок 6 Структура таблицы Manage

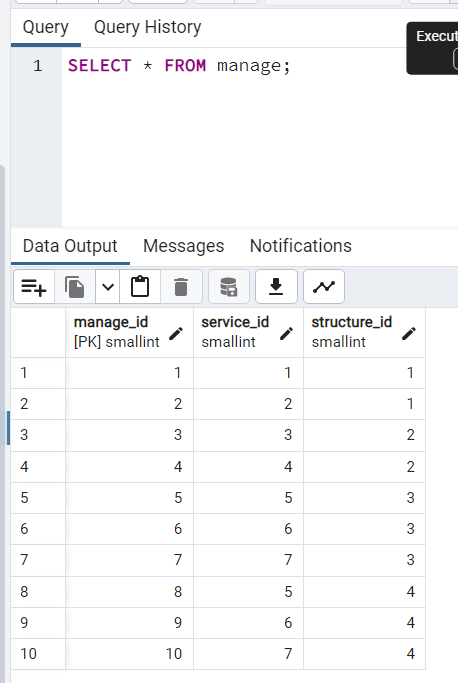
Данная таблица представляет связку двух таблиц. Она соединяет услуги с местом, в которых данная услуга предоставляется.

Рисунок 7 Наполнение таблицы Manage

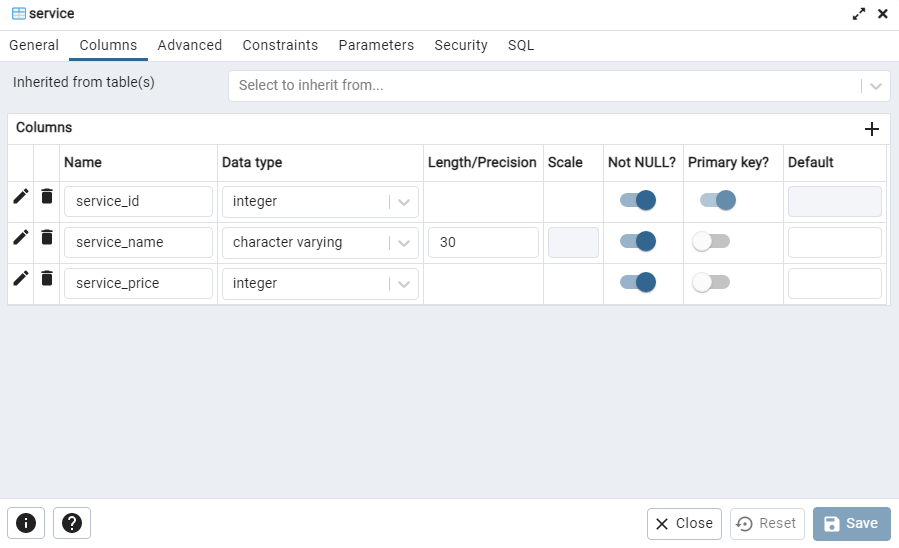
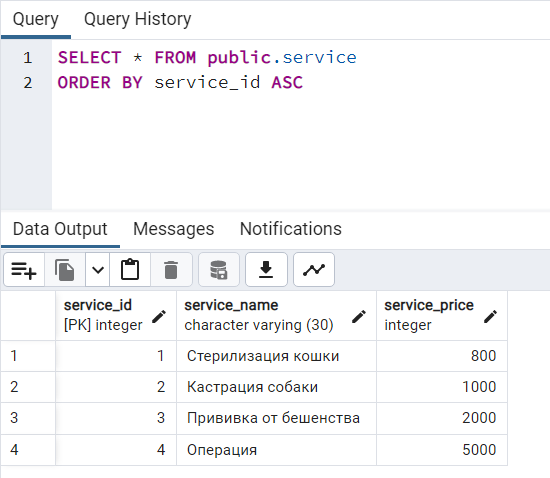
* Таблица «Service»

Рисунок 8 Структура таблицы Service

Данная таблица хранит информацию об услугах, а именно название и цену.

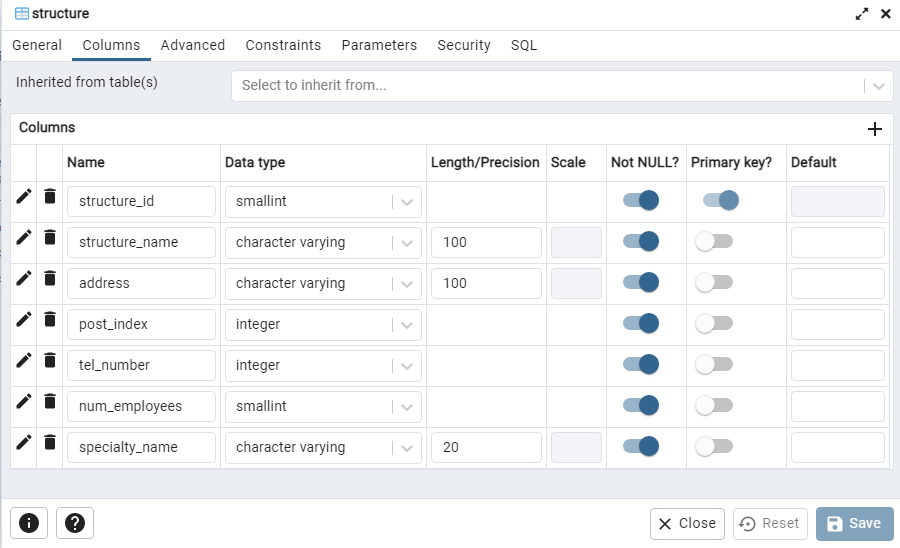
Таблица «Structure»

Рисунок 9 Наполнение таблицы Service

Рисунок 10 Структура таблицы Structure

Данная таблица хранит информацию о заведениях.

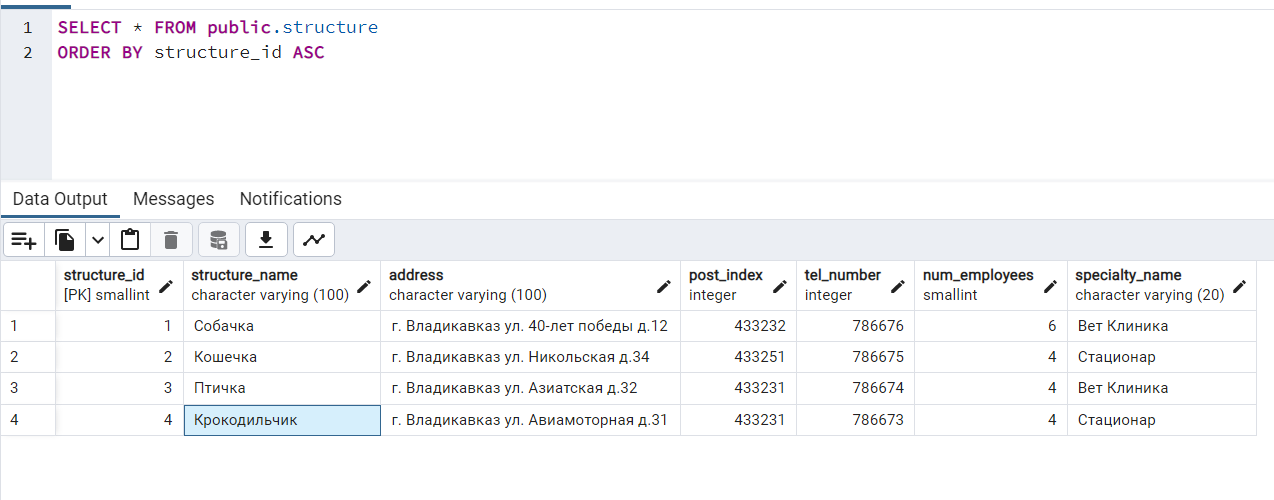


Рисунок 11 Наполнение таблицы Structure

* Таблица «User»

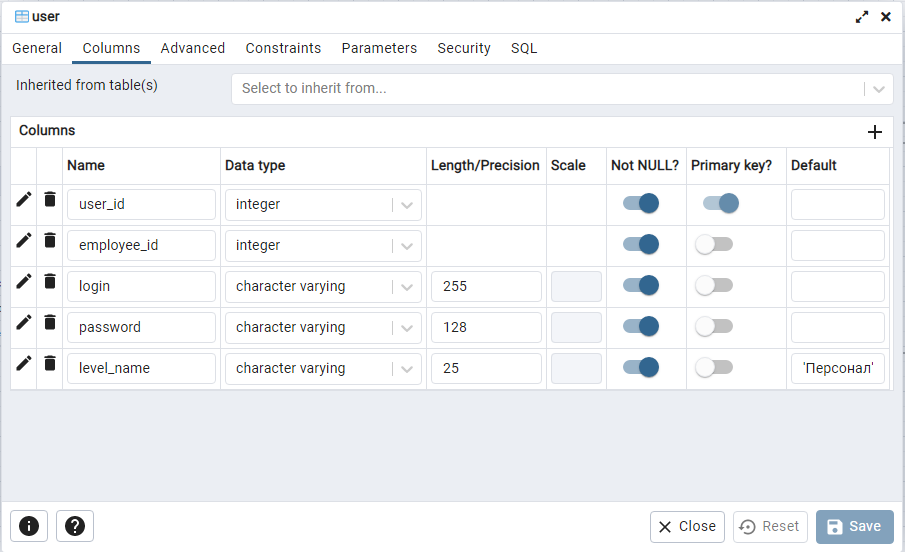


Рисунок 12 Структура таблицы User

Данная таблица хранит информацию о конкретных пользователях системы т.е. хранит логины и пароли, а также уровень доступа у данного пользователя. Заметим, что password занимает 128 символов, это сделано специально, ведь в этой колонке хранится пароль захешированный при помощи алгоритма sha-512 с солью.

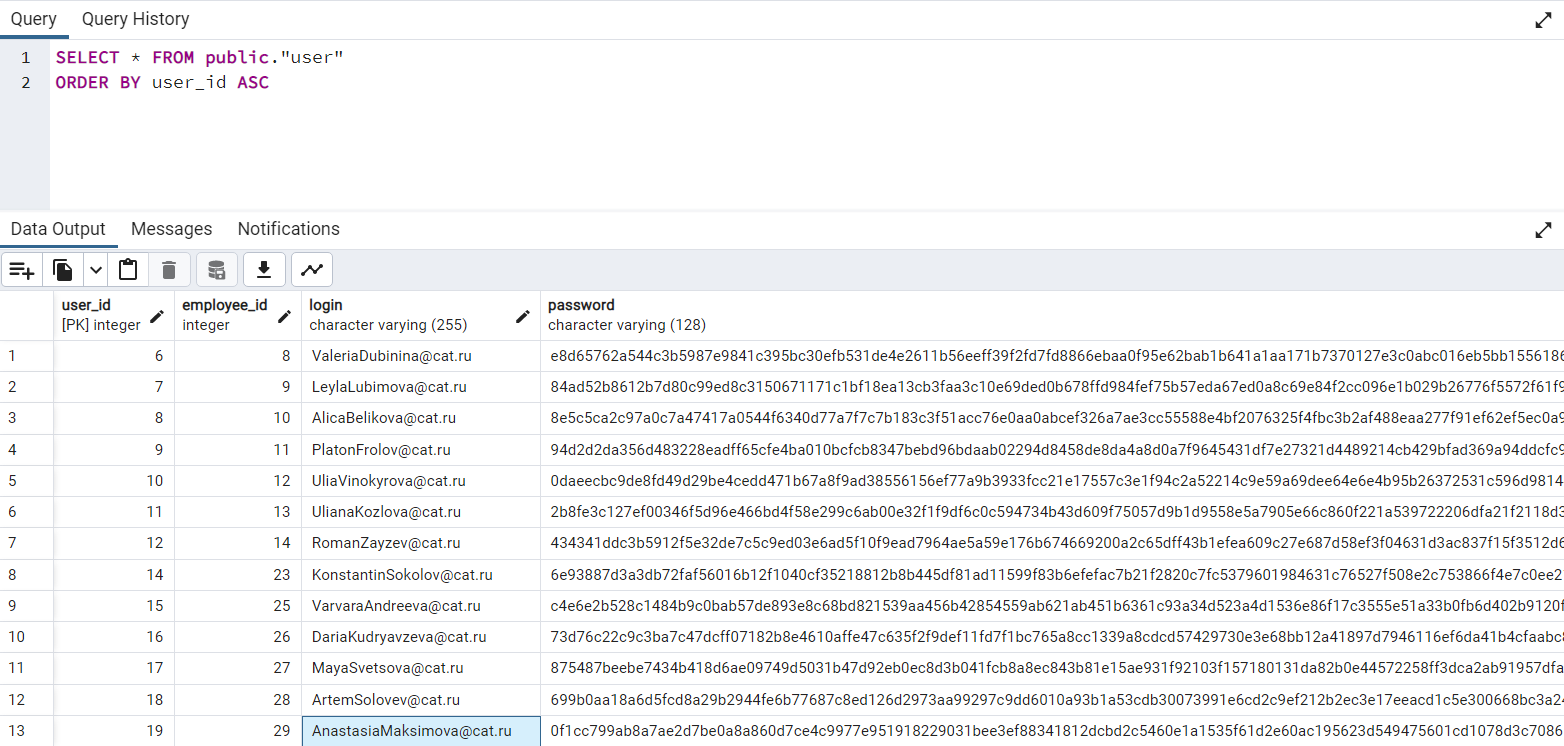


Рисунок 13 Наполнение таблицы User

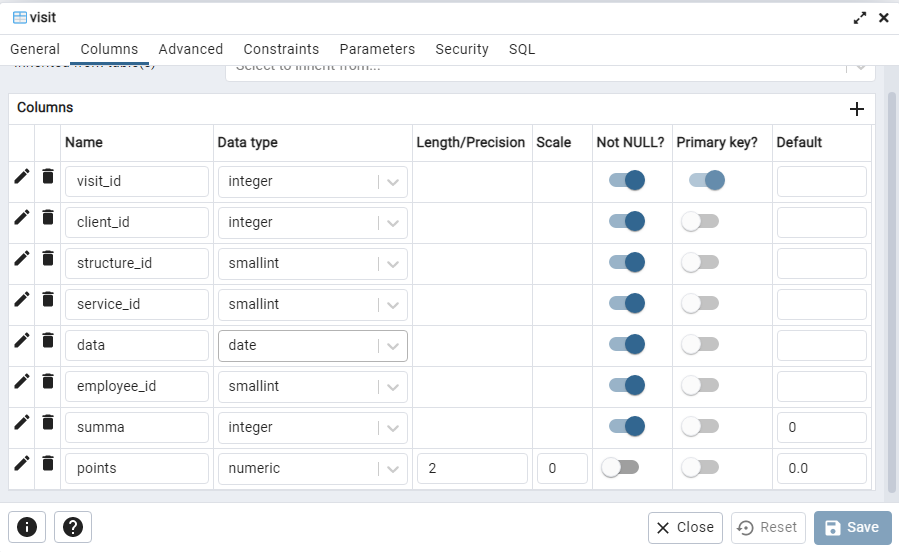
* Таблица «Visit»

Рисунок 14 Структура таблицы Visit

Данная таблица хранит информацию о посещениях.

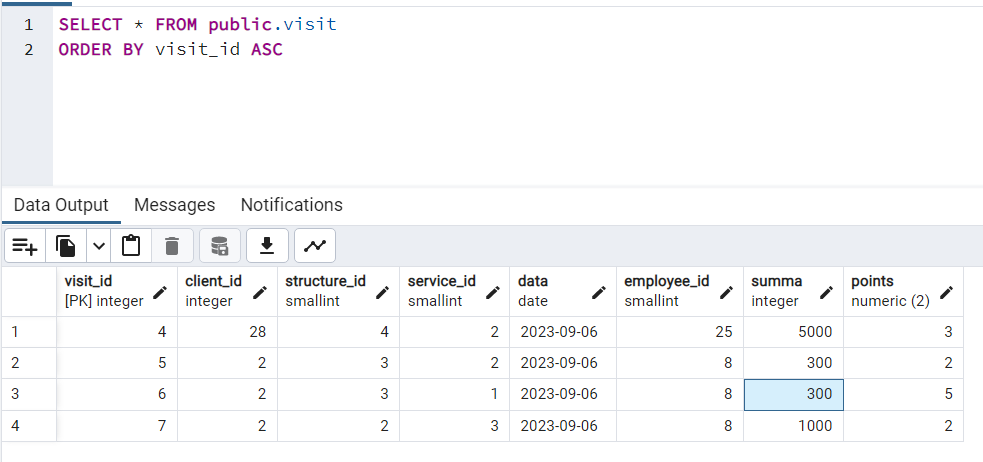


Рисунок 15 Наполнение таблицы Visit

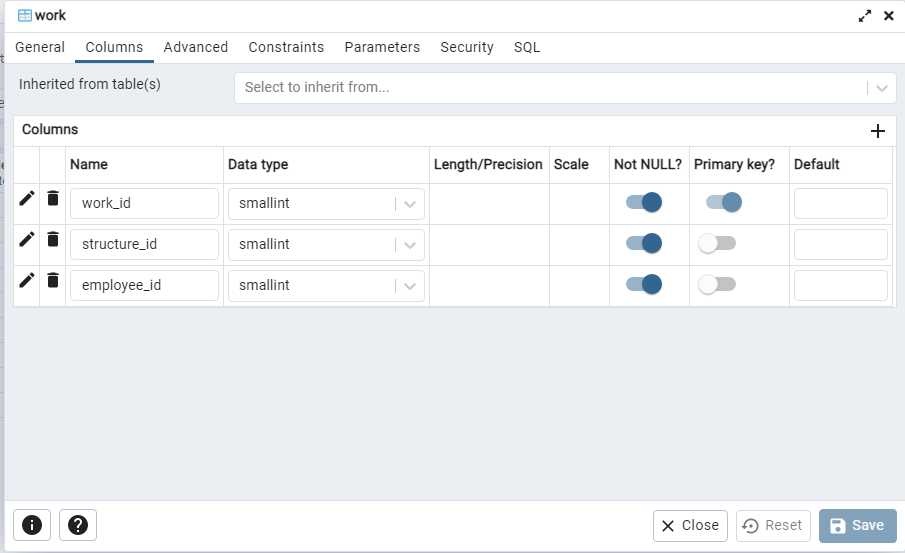
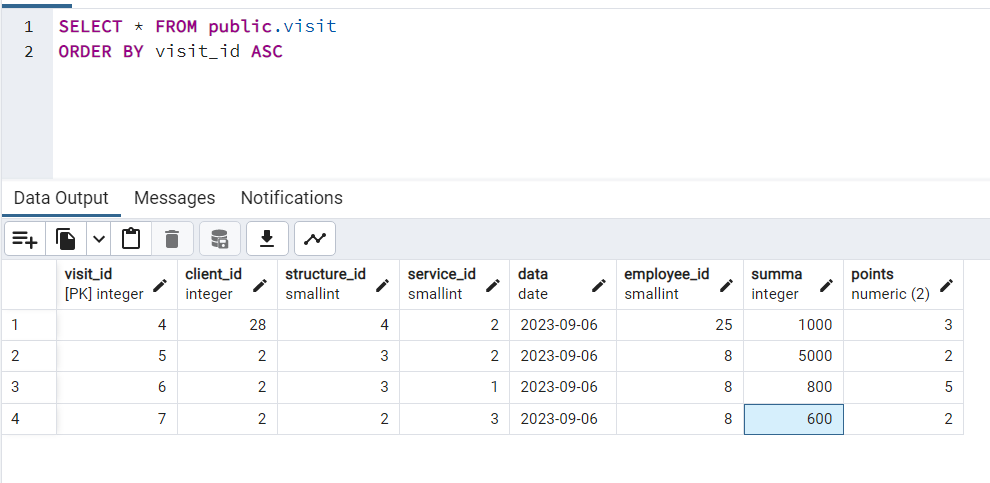
* Таблица «Work»

Рисунок 16 Структура таблицы Work

Данная таблица служит связкой для таблиц «Structure» и «Employee». Она необходимо чтобы понимать, в каком заведении работает конкретный сотрудник.

Рисунок 17 Наполнение таблицы Work



**Задание 3 - 4**

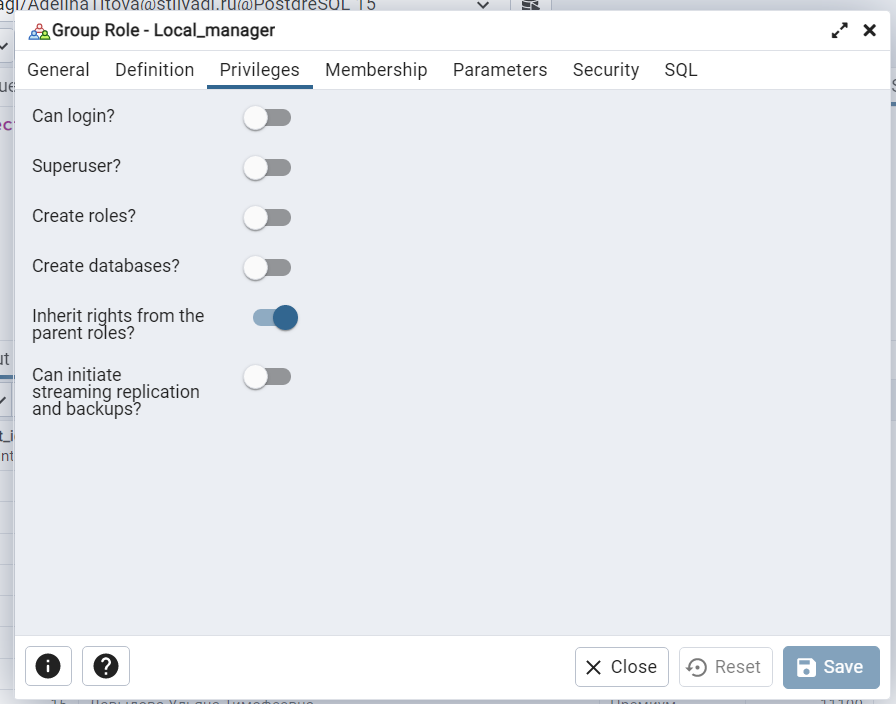
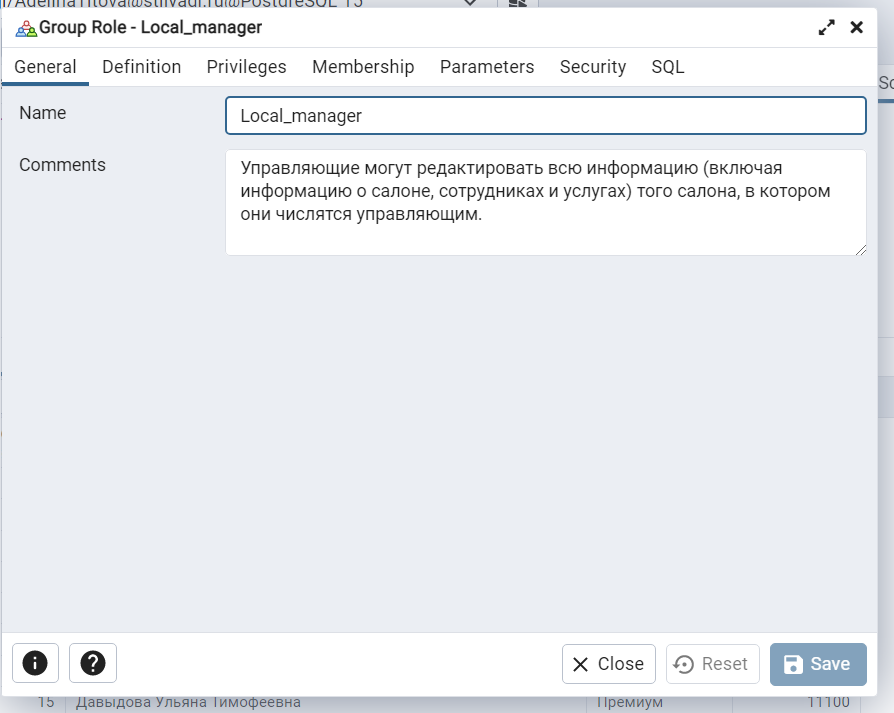
Создадим роль Управляющего(Local\_manager).

Рисунок 18 Создание учетки Управляющего1

Рисунок 19 Создание учетки Управляющего 2

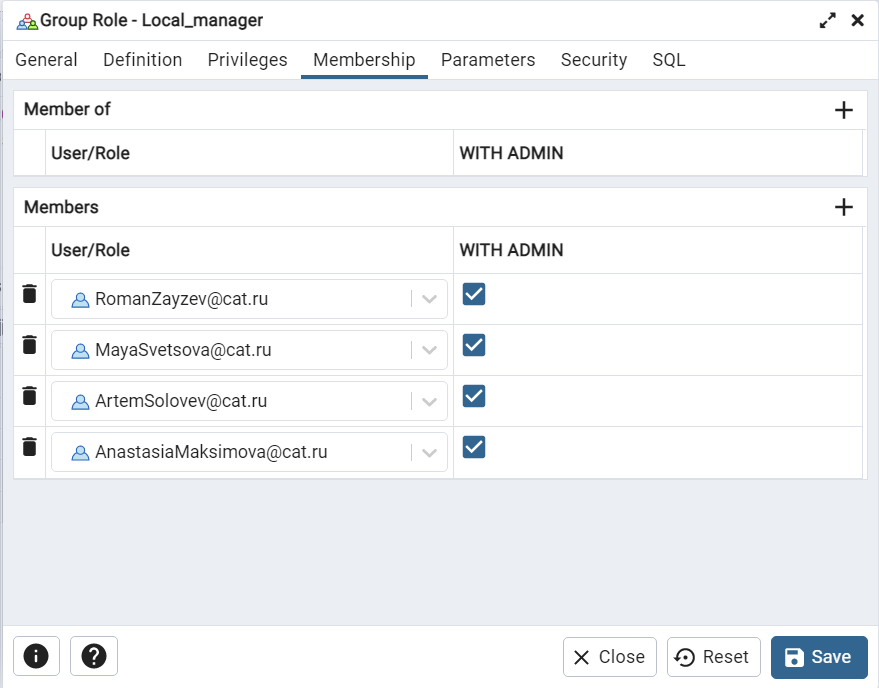


Рисунок 20 Создание учетки Управляющего 3

Пропишем правила policy для управляющего. В нашем случае управляющий – это Local\_manager. Нам необходимо прописать 2 правила для редактирования информации о салоне, в котором работает данный управляющий, а также о его подчиненных.

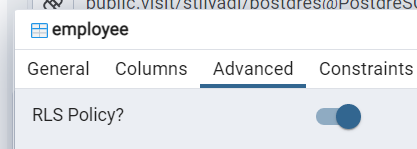
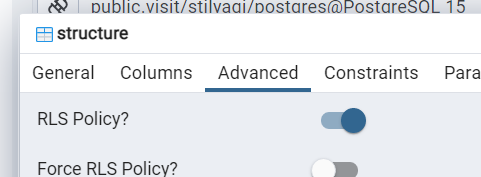
1. Для редактирования информации о своих подчиненных

create policy edit\_employee on employee TO "Local\_manager" using(employee\_id in (select employee\_id from work where structure\_id = (select structure\_id from work join "user" using(employee\_id) where login = current\_user))) ;

1. Для редактирования информации о своем заведении

create policy edit\_structure on structure TO "Local\_manager" using(structure\_id = (select structure\_id from work join "user" using(employee\_id) where login = current\_user));

После того как policy готовы необходимо включить использование правил



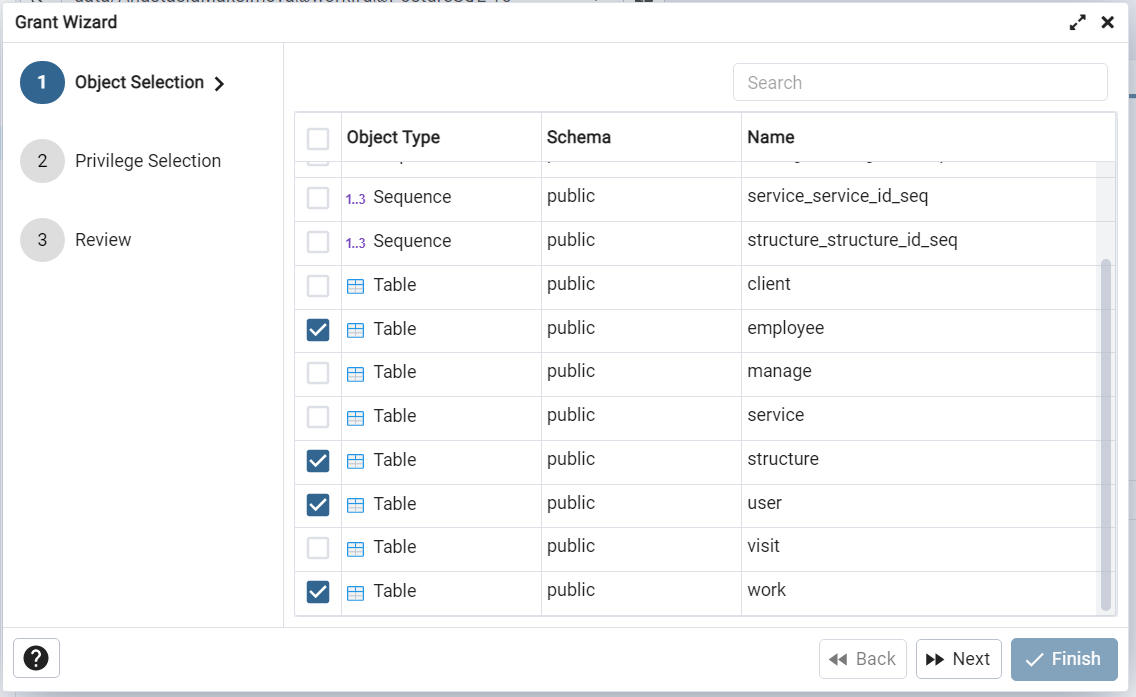
После создания policy необходимо при помощи Grant Wizard выдать ролям, созданным в 3-ем пункте правила.

Рисунок 21 Grand Wizard Управляющего 1

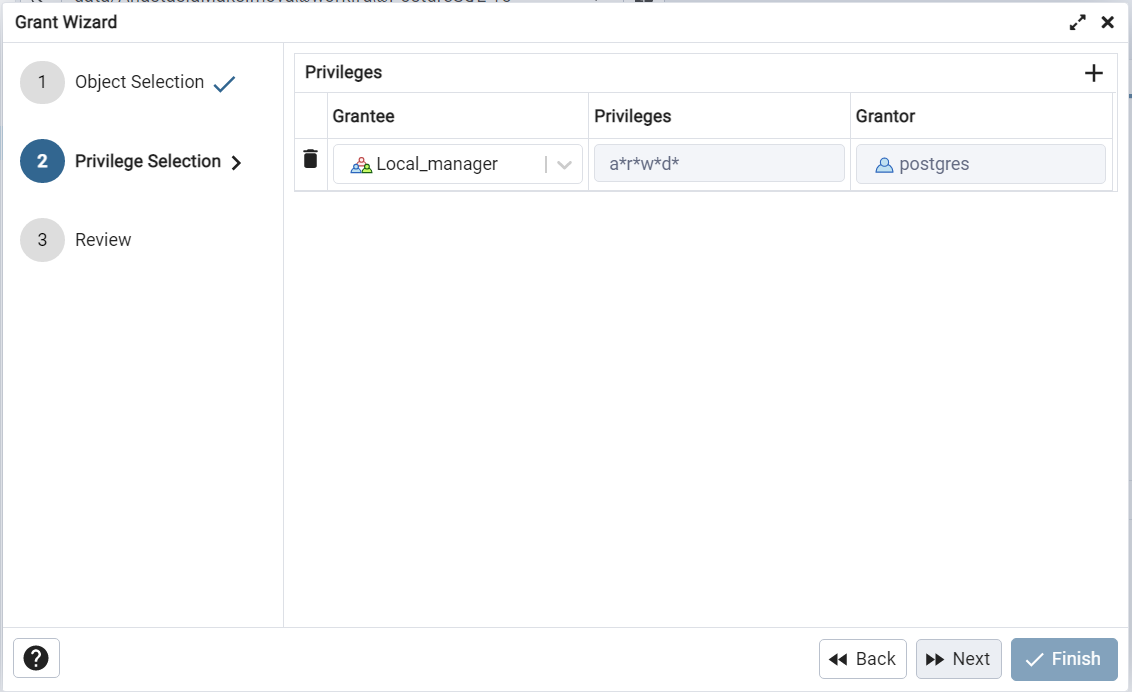


Рисунок 22 Grand Wizard Управляющего 2

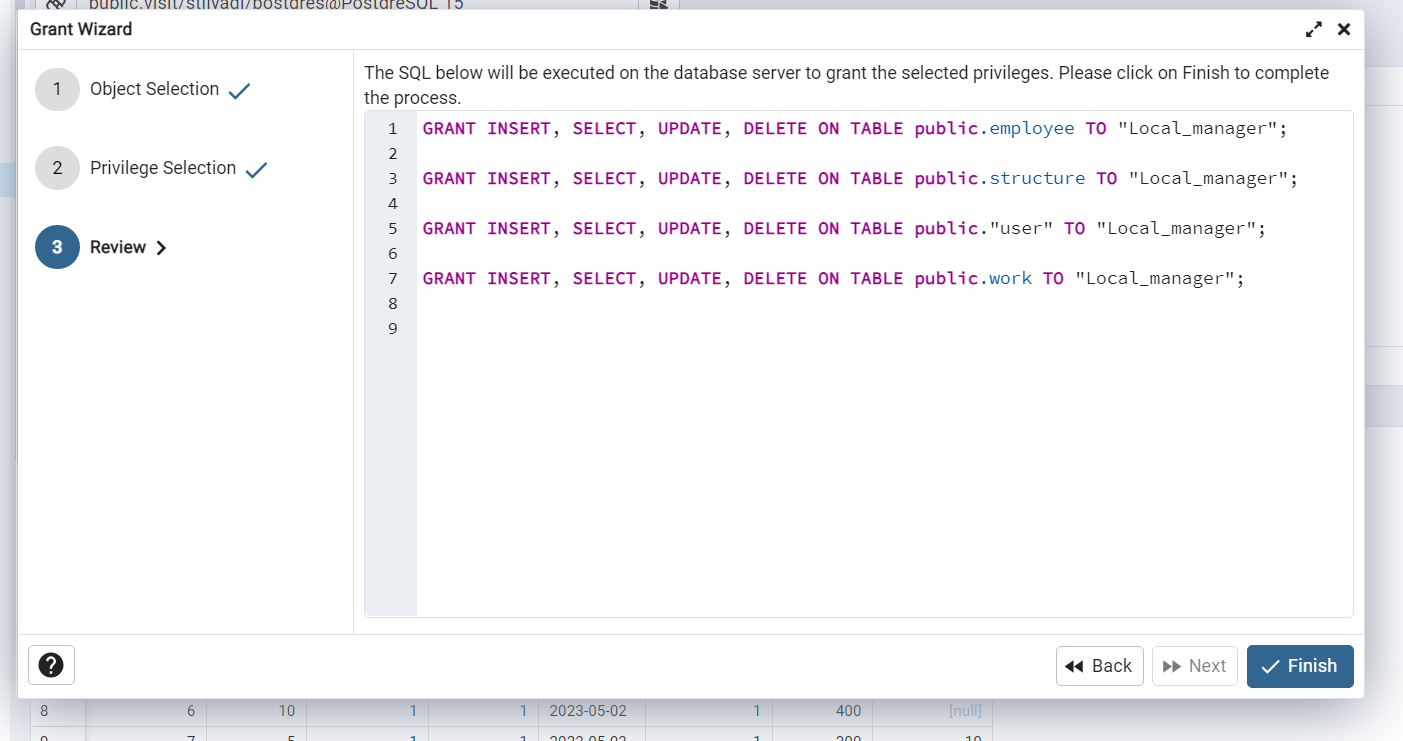


Рисунок 23 Grand Wizard Управляющего 3

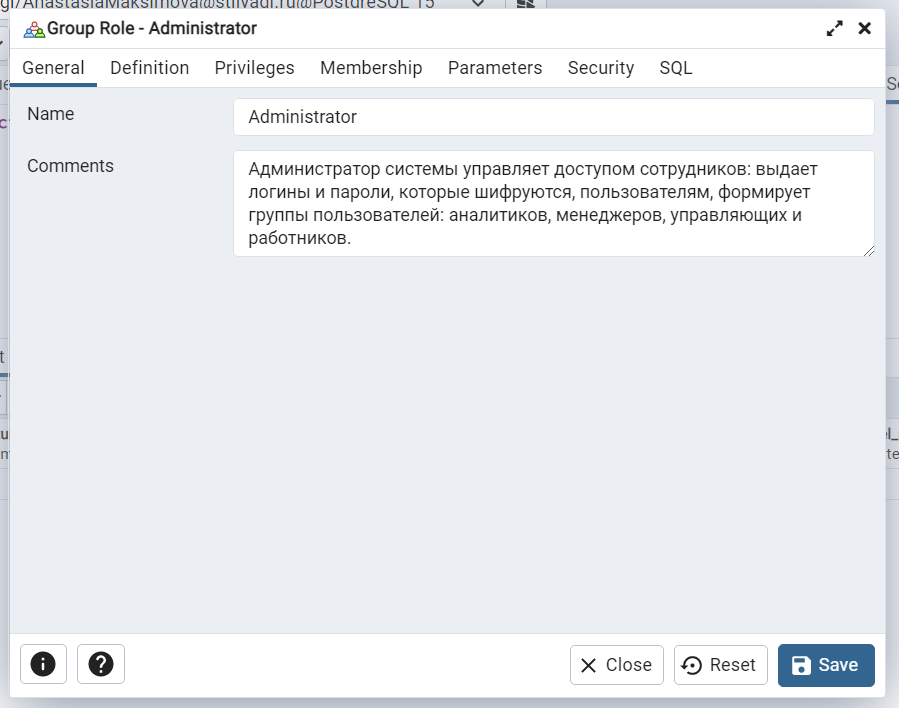
Создадим роль Администратора (Administrator)

Рисунок 24 Создание учетки Админа 1

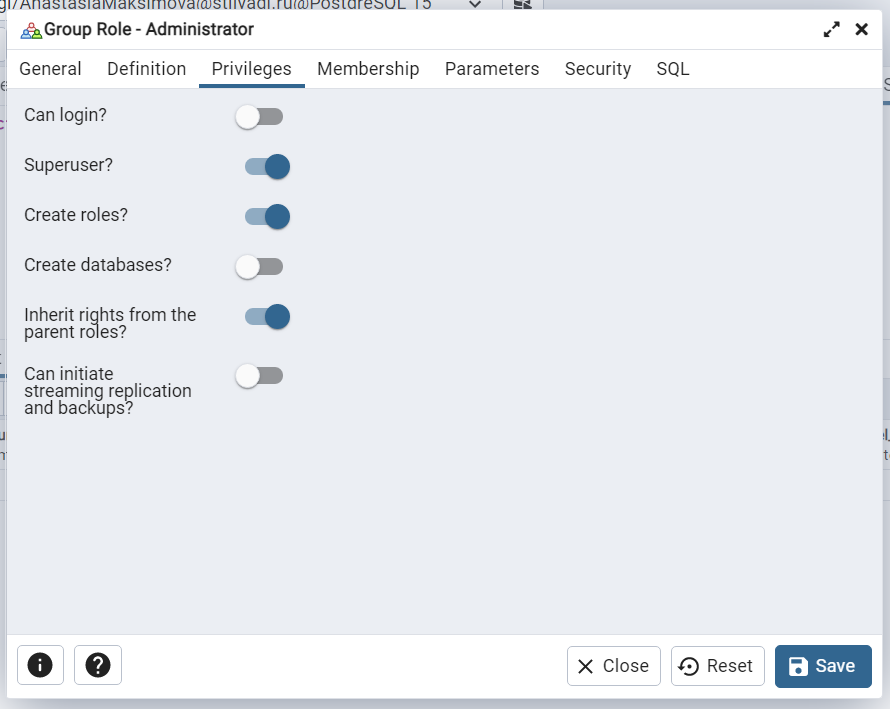


Рисунок 25 Создание учетки Админа 2

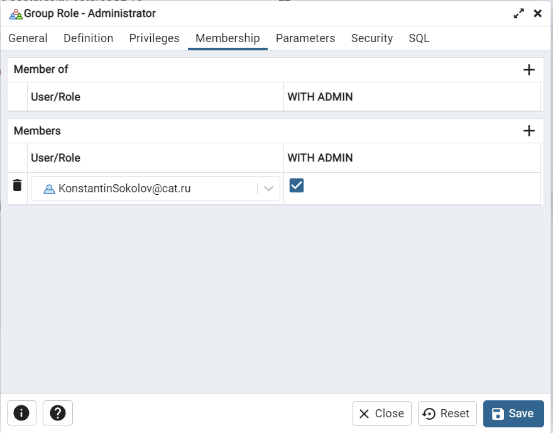


Рисунок 26 Создание учетки Админа 3

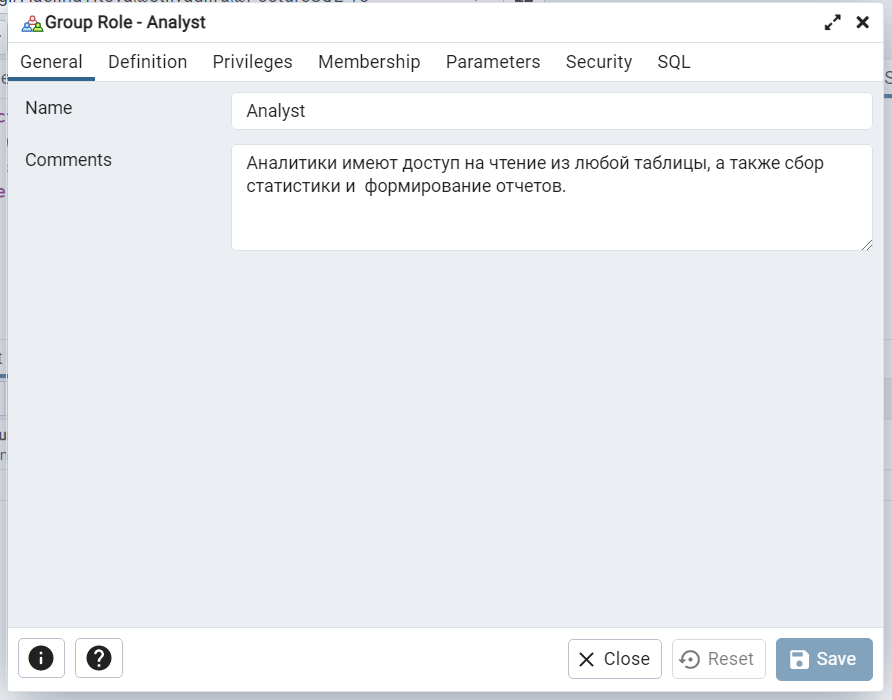
 Создадим роль Аналитика (Analyst).

Рисунок 27 Создание учетки Аналитика 1

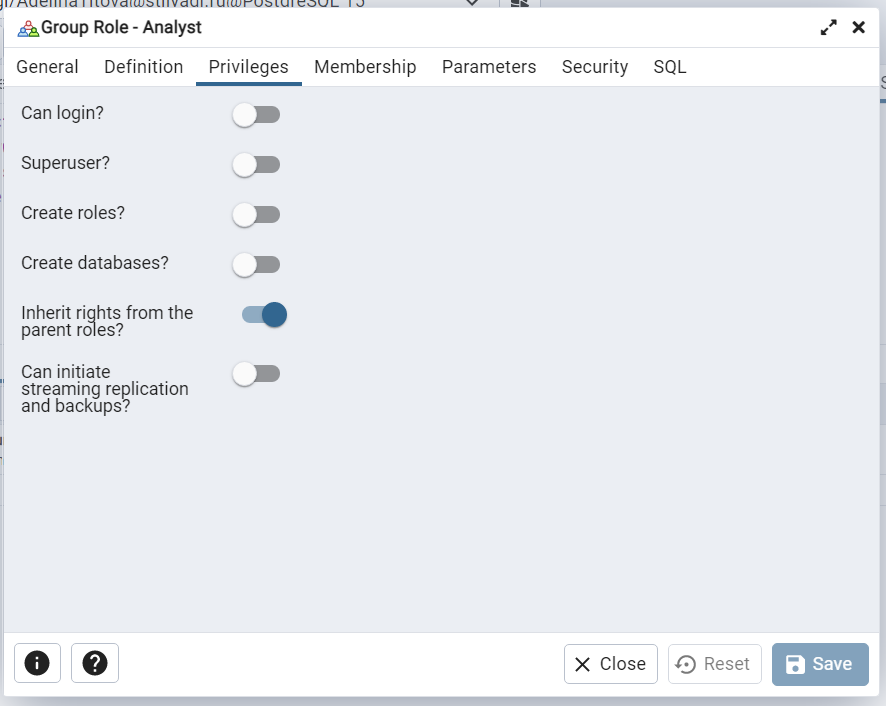


Рисунок 28 Создание учетки Аналитика 2

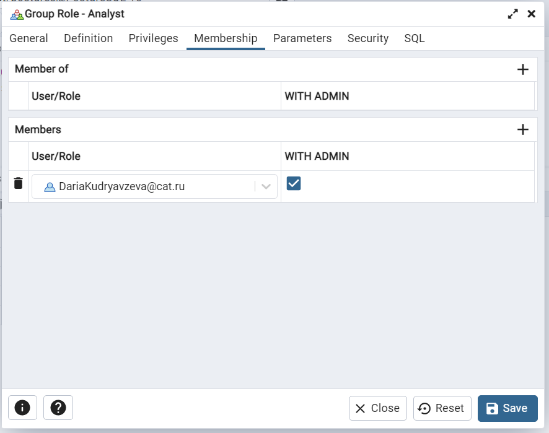


Рисунок 29 Создание учетки Аналитика 3

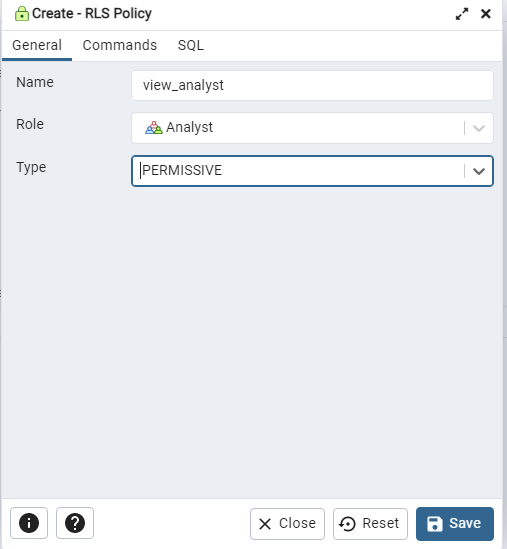
 Сделаем policy для аналитика.

Рисунок 30 Создание policy 1

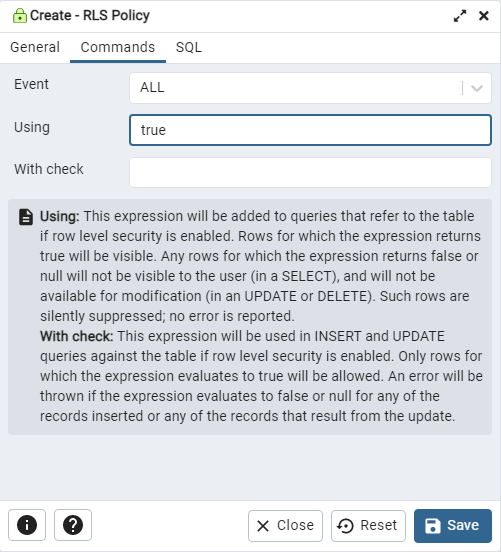


Рисунок 31 Создание policy 2

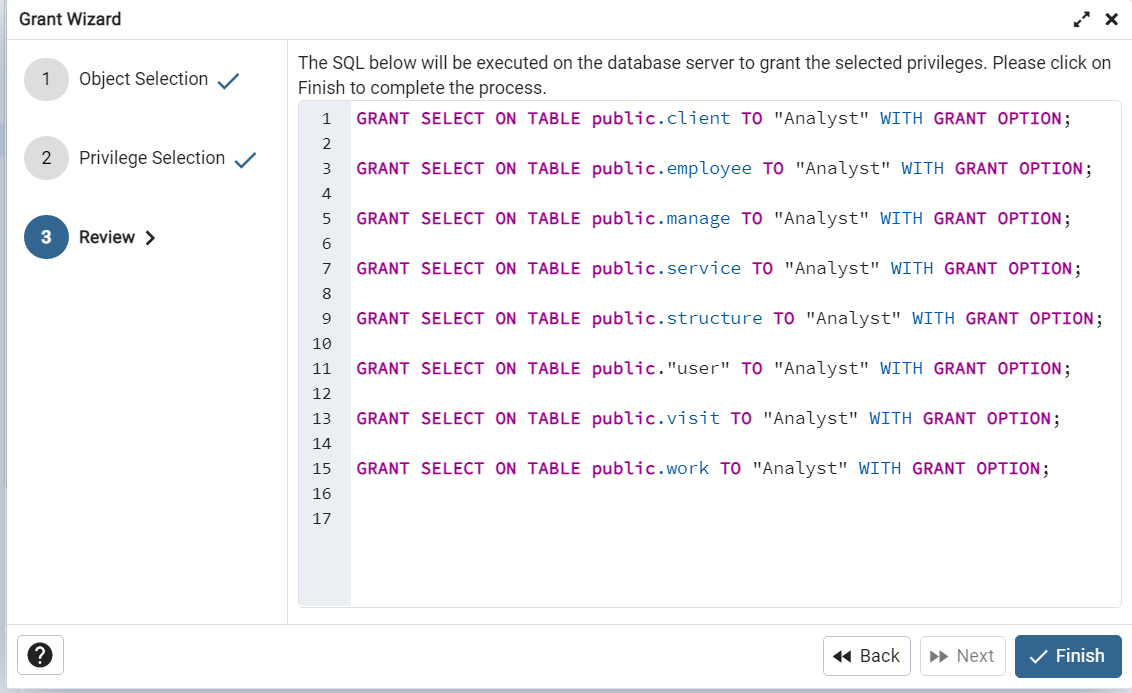
 Эти правила необходимы, чтобы просматривать таблицу employee и structure не имея прав Administrator. Далее добавим полномочий при помощи Grant Wizard на просмотр всех таблиц.

Рисунок 32 Grand Wizard Аналитика

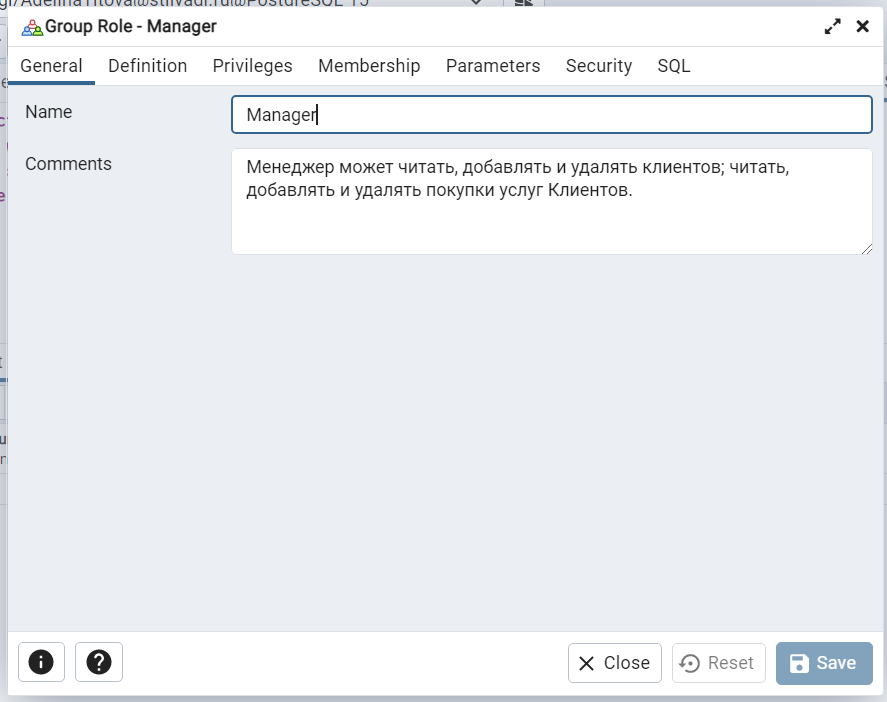
Создадим роль Менеджер (Manager).

Рисунок 33 Создание учетки Манеджера 1

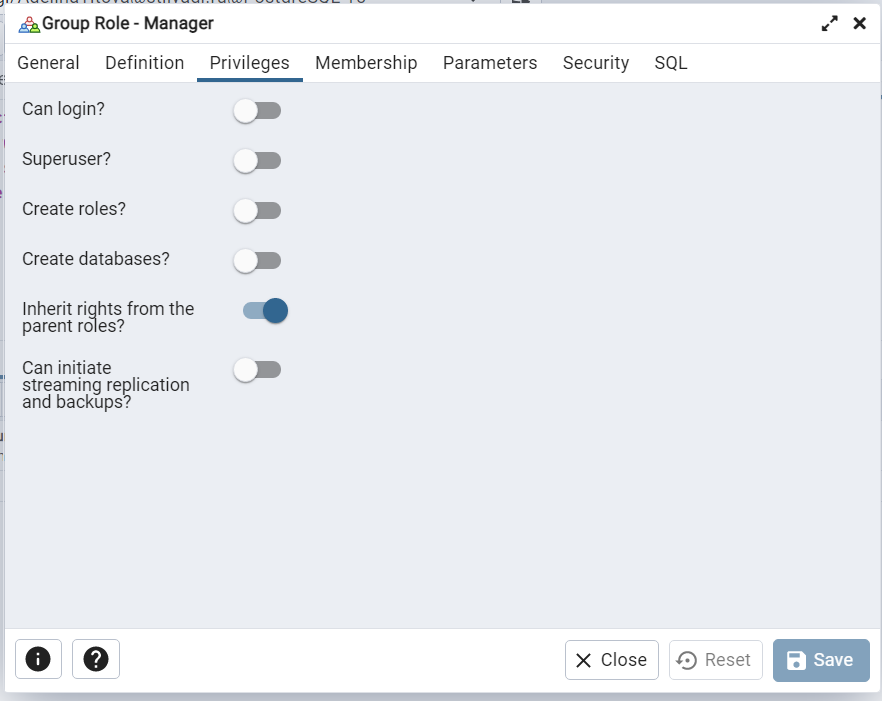


Рисунок 34 Создание учетки Менеджера 2

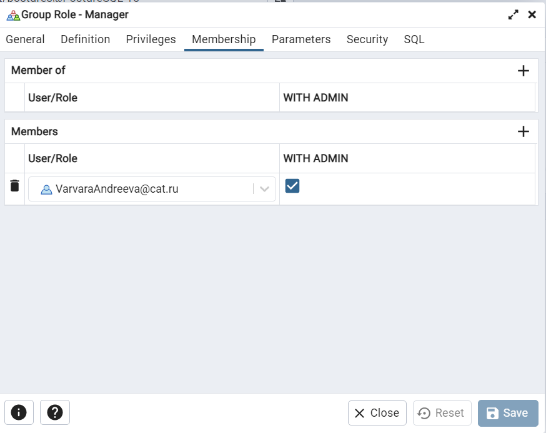


Рисунок 35 Создание учетки Менеджера 3

По заданию они могут изменять таблицу visit и таблицу client. Для этого нам необходимо создать новое правило policy чтобы при изменении таблицы visit у нас менялся рейтинг у сотрудника.

create policy ttaatt on employee using(employee\_id in (select employee\_id from work where current\_user = 'VarvaraAndreeva@work.ru'));

Далее необходимо выдать право на чтение, редактирование, удаление и добавление элементов в таблицах visit и client.

Рисунок 36 Grand Wizard Менеджера

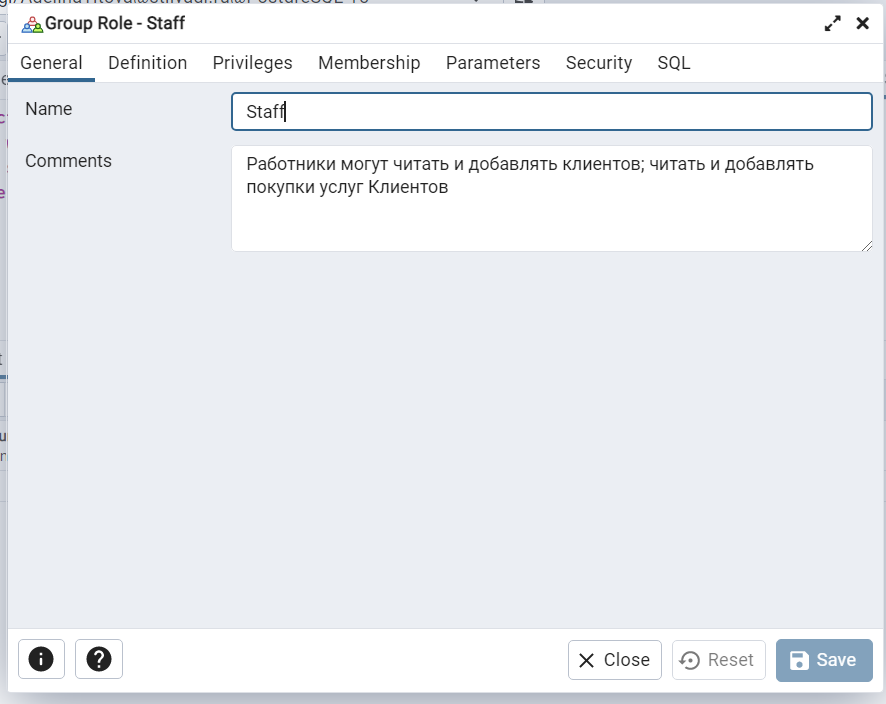
Последний шаг – это разобраться с простыми сотрудниками. Создадим роль Персонала (Staff).

Рисунок 37 Создание учетки Персонала 1

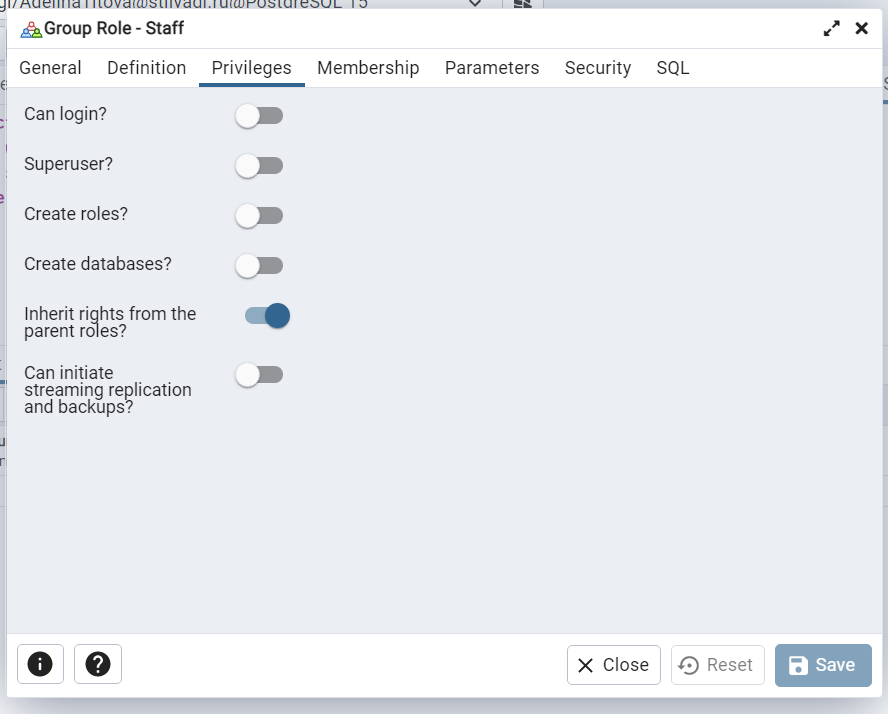


Рисунок 38 Создание учетки Персонала 2

Они, по условию, могут добавлять заказы и клиентов. При помощи Grant Wizard дадим им это право. Необходимо выдать право на просмотр(select) и добавление(insert) в таблицах client и visit



Рисунок 40 Grand Wizard Персонала

**Задание 5**

В нашем случае, необходимо создать 4 триггера, которые будут располагаться на таблице visit.

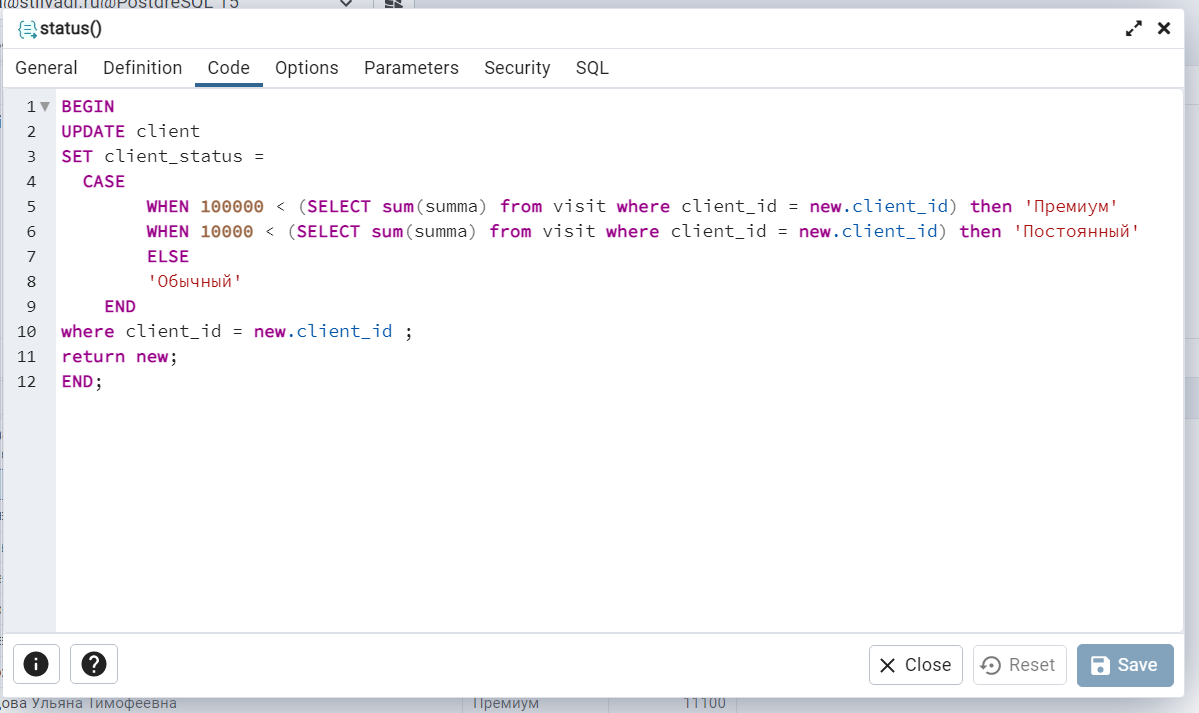
1-ый будет отвечать за статус клиента. В нашем случае если клиент потратил более 100.000 рублей, то он становится «Премиум», если потратил больше 10.000 рублей, то он становится «Постоянным»

Рисунок 41 Триггер №1

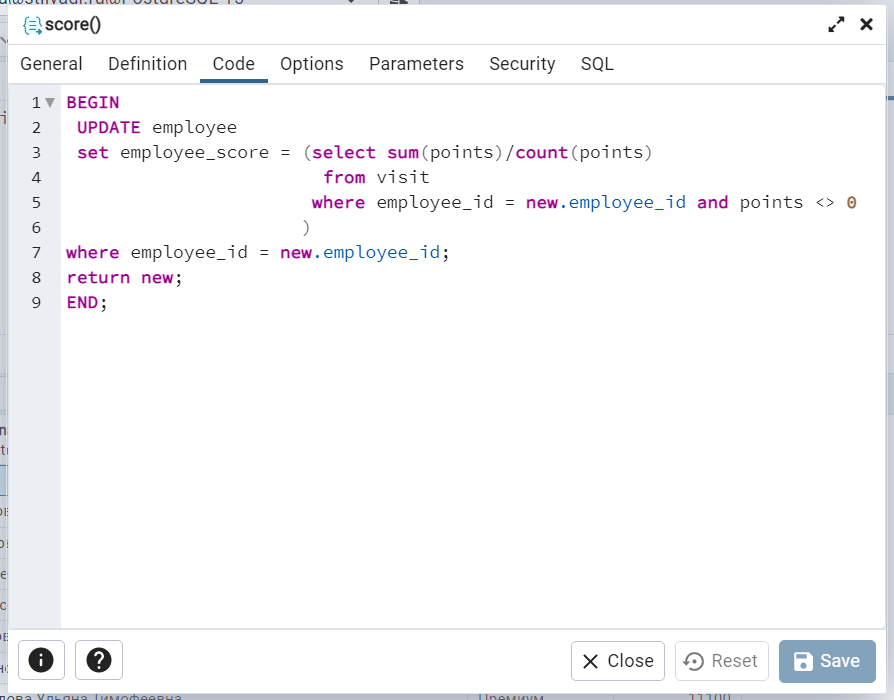
2-ой триггер будет отвечать за рейтинг парикмахеров. Он будет считать сумму всех не нулевых оценок и делить на их количество.

Рисунок 42 Триггер №2

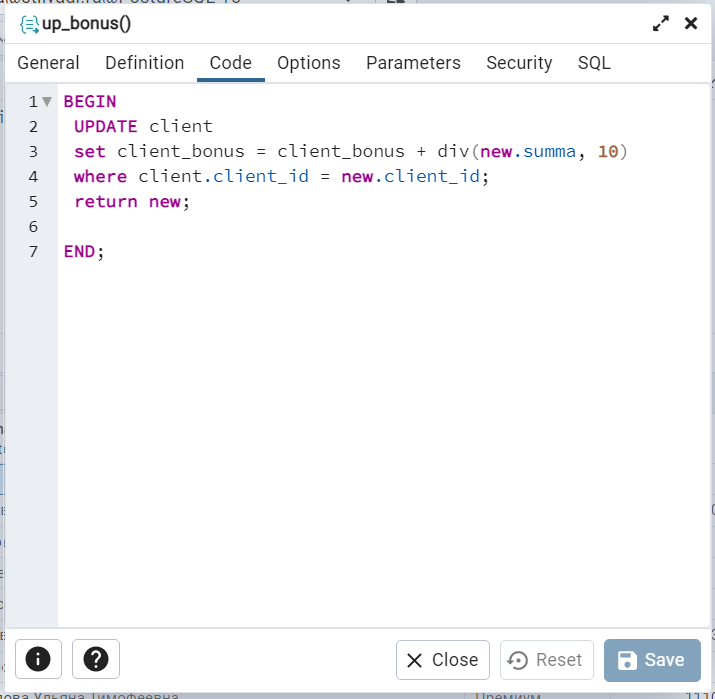
 3-ий и 4-ый триггеры отвечают в целом за одно и тоже. Они считают бонусные рубли для каждого клиента. Бонусная программа - это 10% от суммы заказа. Разница лишь в том, что один работает на INSERT, а второй работает на UPDATE.

Рисунок 43 Триггер №3

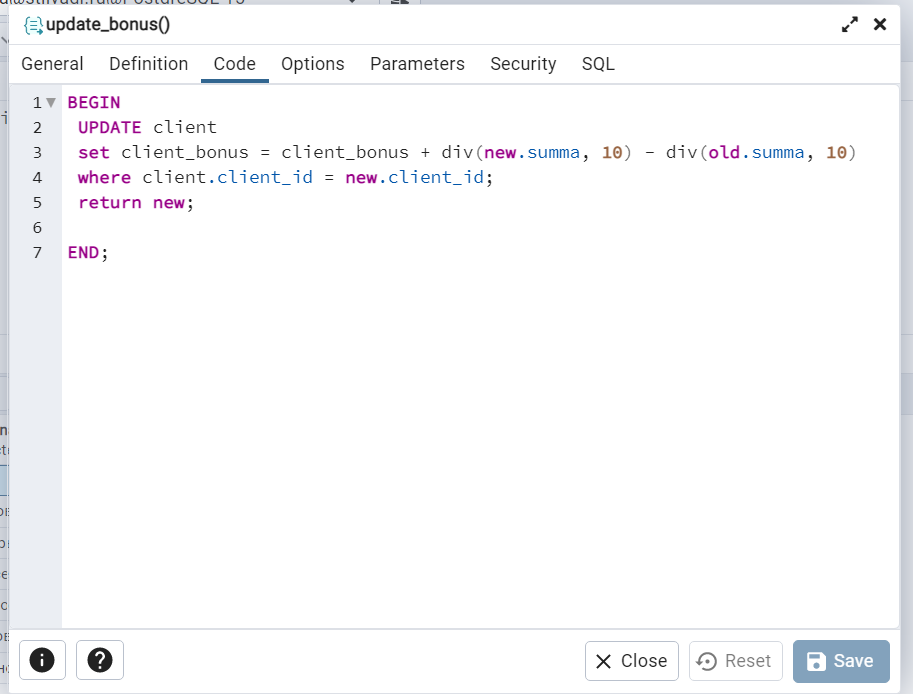
 **Задание 6**

Рисунок 44 Триггер №4

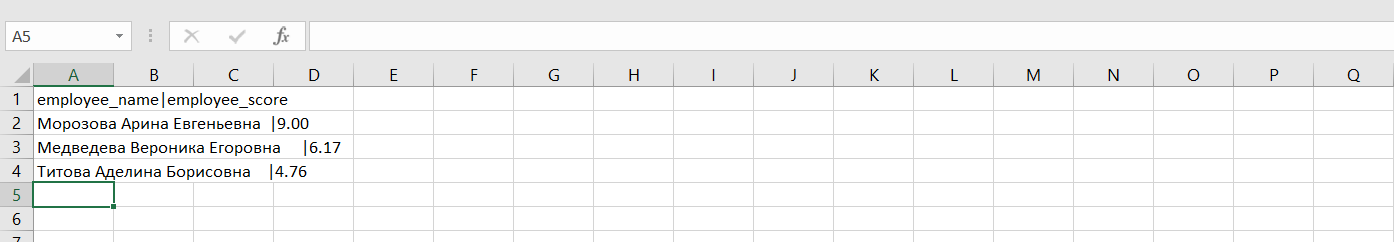
 Отчет о лучших сотрудниках месяца.

Рисунок 45 Запрос о сотрудниках месяца

Рисунок 46 Отчет о сотрудниках месяца

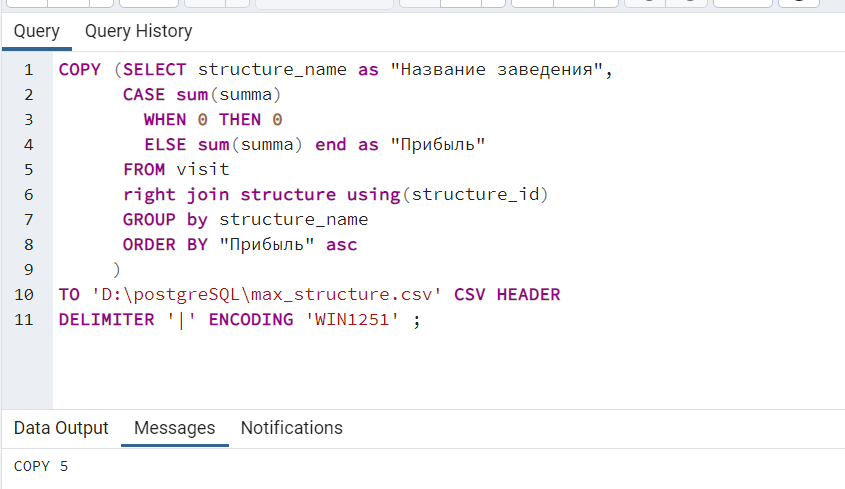
 Отчет о прибыли каждого учреждения.

Рисунок 47 Запрос о доходе каждого заведения

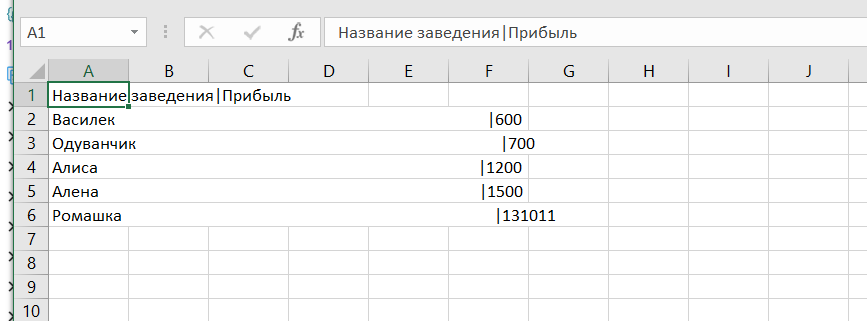


Рисунок 48 Отчет о доходе каждого заведения

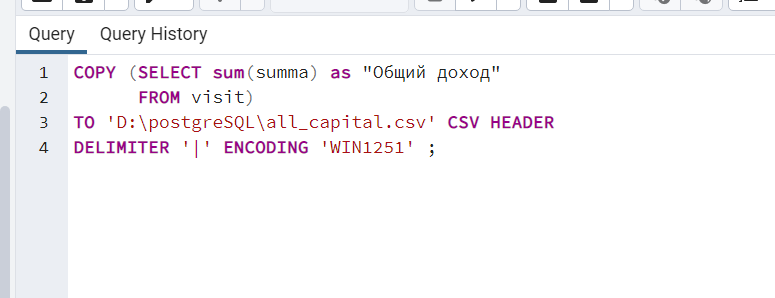
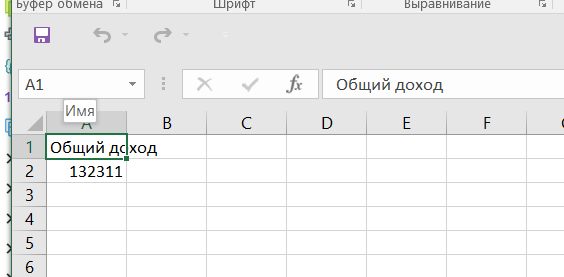
Общий доход заведений.

Рисунок 50 Отчет об общем доходе

Рисунок 49 Запрос об общем доходе

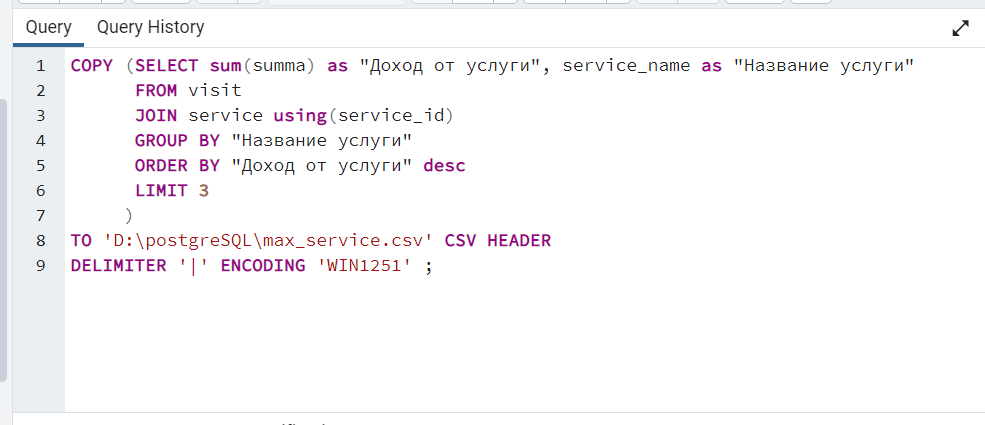
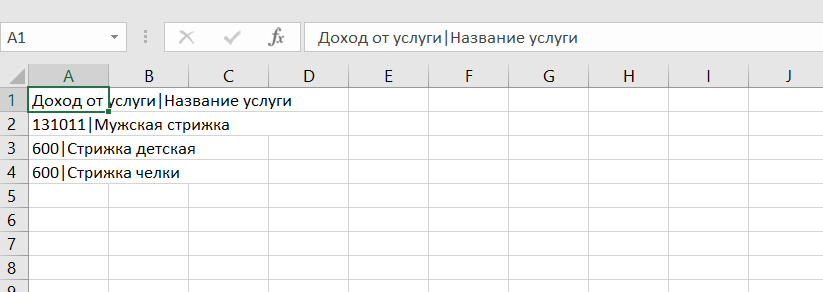
 Отчет о самых прибыльных услугах

Рисунок 52 Отчет о самых прибыльных услугах

Рисунок 51 Запрос о самых прибыльных услугах

 Отчет о самых неприбыльных услугах

Рисунок 53 Запрос о самых неприбыльных услугах

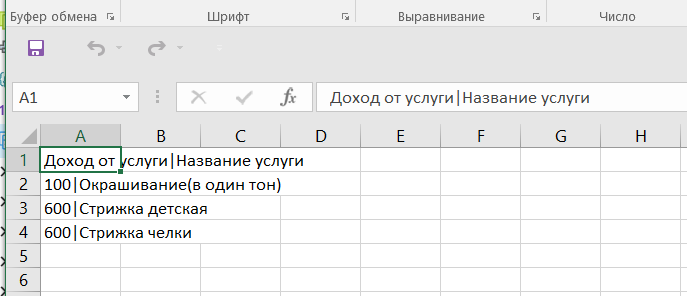


Рисунок 54 Отчет о самых неприбыльных услугах

Использованные ресурсы

* <https://poformule.ru/generatory/generator-telefonnykh-nomerov>
* <https://randomus.ru/name?type=0&sex=10&count=10>
* <https://postgrespro.ru/docs/postgresql/13/sql>
* <https://stackoverflow.com>