



Промышленные СУБД

Лабораторная работа №2

Лабораторная работа №2

Система безопасности PostgreSQL

Цель: научить использовать системные хранимые процедуры для управления именами входа PostgreSQL и пользователями баз данных, а также разрешать и запрещать выполнение определенных действий некоторому пользователю.

Требования к отчету: по результатам работы представить отчёт со скриншотами, содержащими SQL-команды и результаты их выполнения для каждой задачи из раздела «Самостоятельная работа».

Задание 1. Подключитесь к серверу WS0481 (либо же зайдите на какой-либо существующий сервер на вашем собственном компьютере)

Указания к выполнению:

1. Запустите pgAdmin 4 через меню Пуск – pgAdmin 4
2. Создайте новый сервер: нажмите кнопку «Add New Server» укажите Имя для сервера – WS0481. Во вкладке «Connection» укажите адрес в поле «Host name/address» – localhost, установите пароль и нажмите кнопку «Save».

Задание 2. Определите список ролей сервера.

ВАЖНО! Роли и пользователи в PostgreSQL синонимичны с недавних пор.

Указания к выполнению:

1. В pgAdmin 4 откройте ветвь Servers → <название вашего сервера> → Login/Group Roles, либо выполните данный запрос (рис. 1):

```
SELECT rolname FROM pg_roles;
```

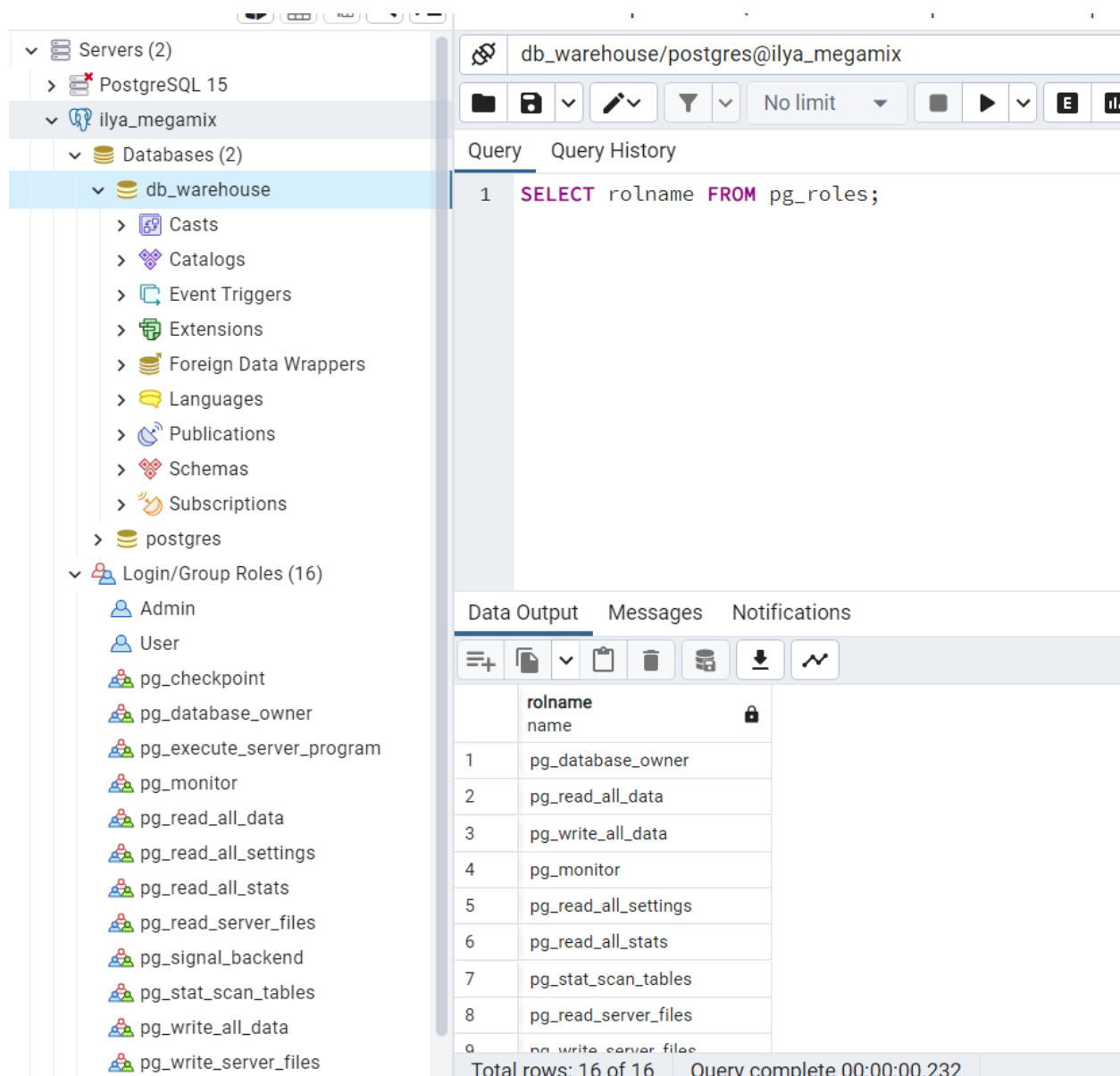


Рис. 1. Просмотр ролей

Задание 3. Создайте и настройте новую учетную запись *temp_user* для входа в PostgreSQL.

Указания к выполнению:

1. Для добавления учетной записи используйте либо Create → Login/Group Role (, либо следующую команду (рис. 2):

`CREATE USER temp_user PASSWORD '1234';`

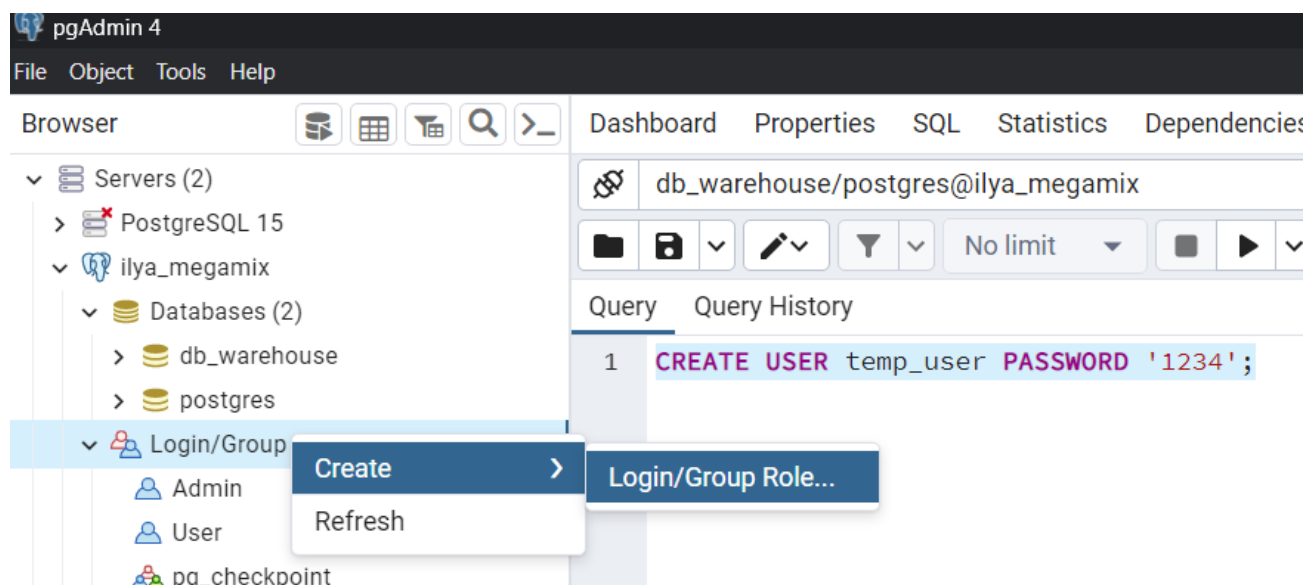


Рис. 2. Создание пользователя

При создании пользователя через интерфейс важно отметить, чтобы он мог “логиниться” (рис. 3), иначе пользователь будет считаться ролью, хотя формально они являются одним и тем же.

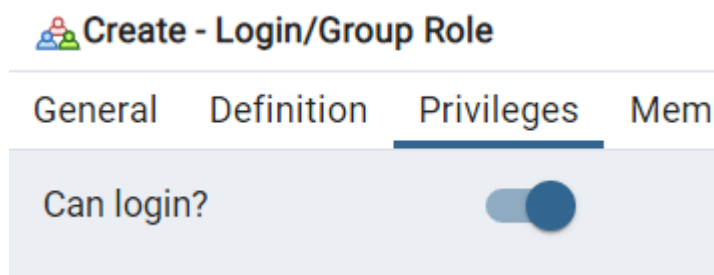


Рис. 3. Возможность “логиниться”

2. Теперь войдите на сервер под новым пользователем (рис. 4-5). Как мы видим на рисунке 6, у него по умолчанию нет прав создавать новых пользователей.

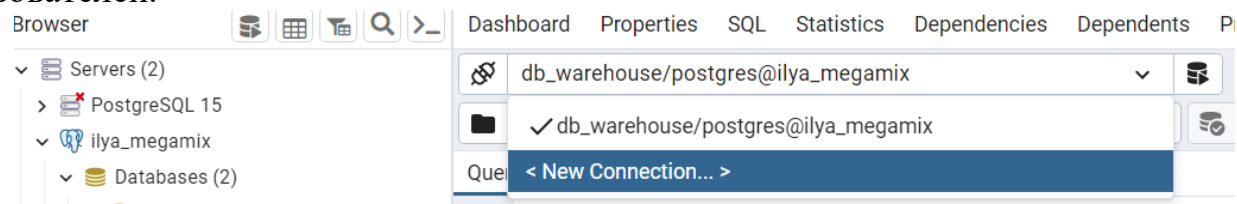
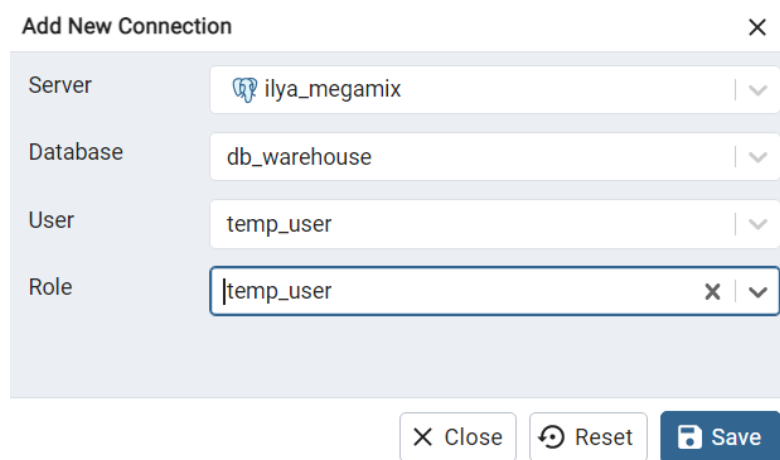


Рис. 4. Установка нового соединения



Add New Connection

Server: ilya_megamix

Database: db_warehouse

User: temp_user

Role: temp_user

Close Reset Save

Рис. 5. Вход под пользователем temp_user

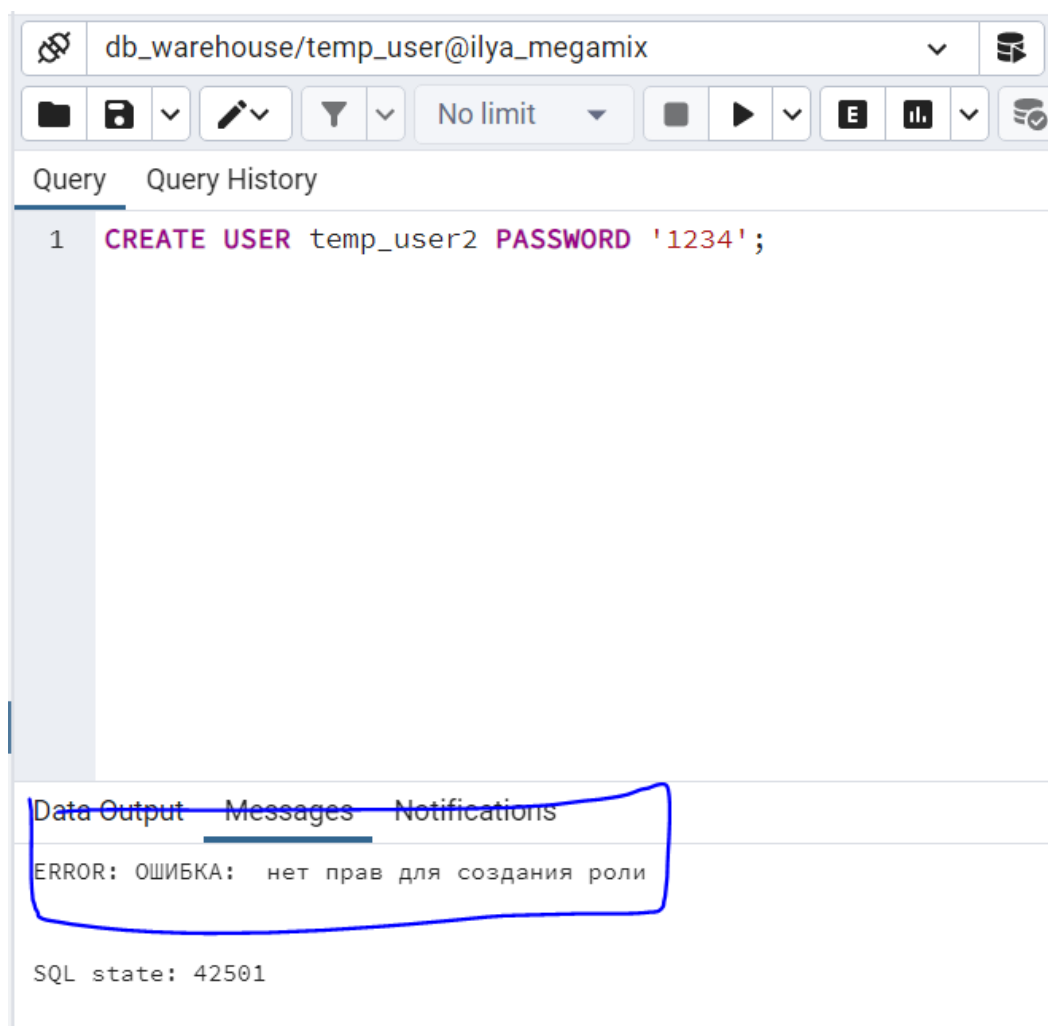


Рис. 6. Пользователю temp_user отказано в доступе

3. Зайдите снова под учетной записью админа. Это понадобится для того, чтобы дать temp_user какие-либо права.

4. Для присвоения пользователю права из роли, зайдите в свойства и в Membership выберите ему роль (рис. 7), либо используя следующую команду:

`GRANT pg_read_all_data TO temp_user;`

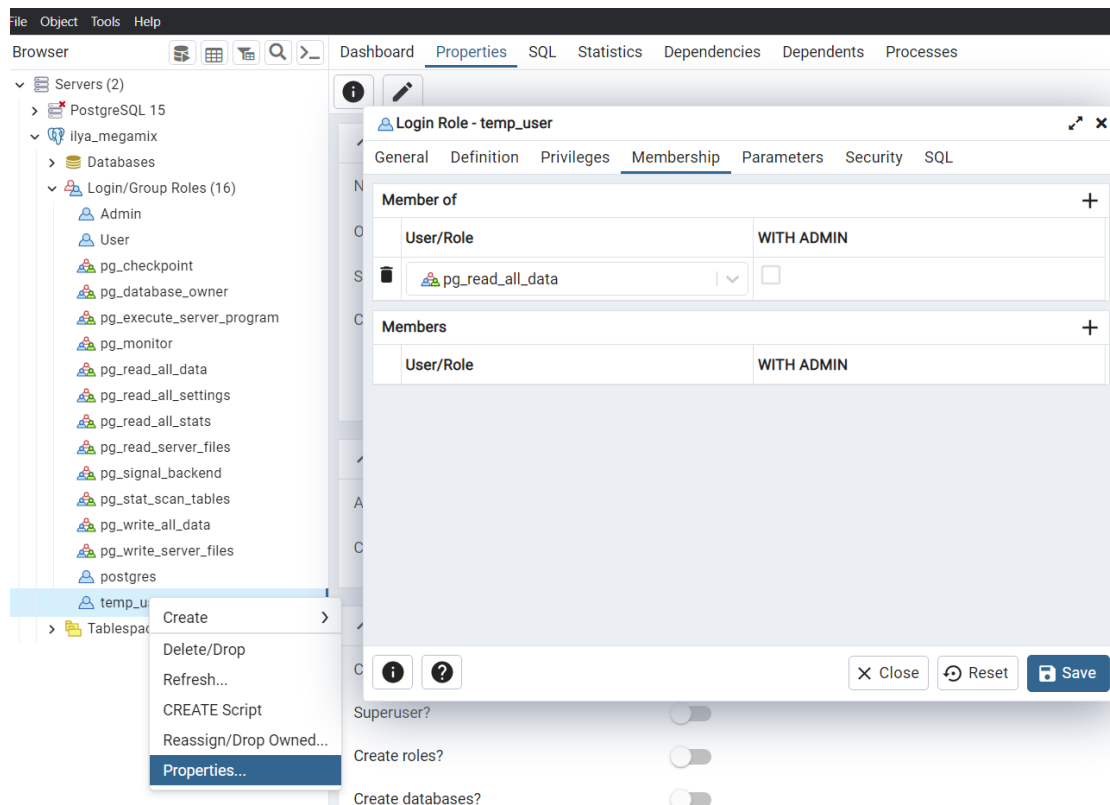


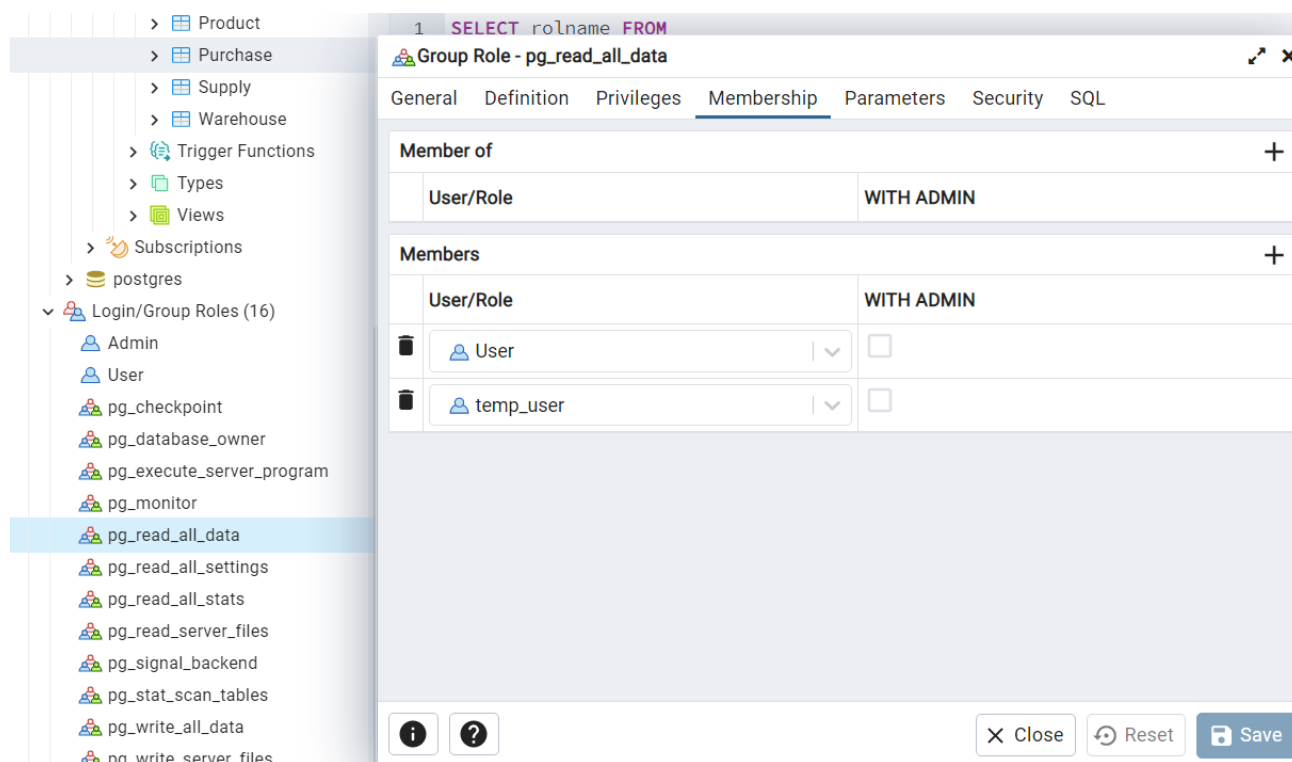
Рис. 7. Присвоение роли пользователю

Задание 4. Определите список ролей базы данных и членов роли *pg_read_all_data*.

Указания к выполнению:

1. Либо посмотрите участников роли в свойствах самой роли (рис. 8), либо введите следующий запрос:

```
SELECT r.rolname as username,r1.rolname as "role"
FROM pg_catalog.pg_roles r JOIN pg_catalog.pg_auth_members m
ON (m.member = r.oid)
JOIN pg_roles r1 ON (m.roleid=r1.oid)
WHERE r1.rolname='pg_read_all_data'
ORDER BY 1;
```

Рис. 8. Список членов роли *pg_read_all_data*

Задание 5. Создайте пользователя *CrippledUser*. Настройте права доступа пользователю *CrippledUser*: предоставьте явным образом право только для выборки из таблицы *TempTable* и обновления только полей *FirstColumn* и *SecondColumn* этой таблицы.

Указания к выполнению:

1. От лица администратора создайте пользователя *CrippledUser*. Задайте ему возможность логиниться и какой-либо пароль.
2. Добавьте таблицу *TempTable* с тремя колонками: *FirstColumn*, *SecondColumn*, *ThirdColumn*. Типы не имеют значение. Добавьте несколько записей (рис. 9).
3. Всё ещё работая от лица админа, даём пользователю *CrippledUser* право просматривать две колонки из таблицы *TempTable*, используя данную команду:

```
GRANT SELECT ("FirstColumn", "SecondColumn") ON public."TempTable" TO "CrippledUser"
```

4. Зайдите от лица *CrippledUser* и введите следующую команду:

```
SELECT * FROM public."TempTable"
```

Система не даст этого сделать. Но если ввести:

```
SELECT "FirstColumn", "SecondColumn" FROM public."TempTable"
```

То система без проблем выдаст данные из этих двух колонок.

	FirstColumn [PK] integer	SecondColumn integer	ThirdColumn integer
1	111	222	333
2	11	22	33
3	1	2	3

Рис. 9. Добавление записей в таблицу

Задание 6. Отмените присвоение роли учетной записи и удалите учетную запись *temp_user*.

Указания к выполнению:

1. Отмена присвоенной пользователю роли может быть выполнена либо в свойствах пользователя (рис. 10), либо с помощью следующей команды:

```
REVOKE pg_read_all_data FROM temp_user
```

2. Для удаления пользователя можно либо нажать правой кнопкой мыши по пользователю и нажать “Удалить”, либо выполнить следующую команду:

```
DROP USER temp_user
```

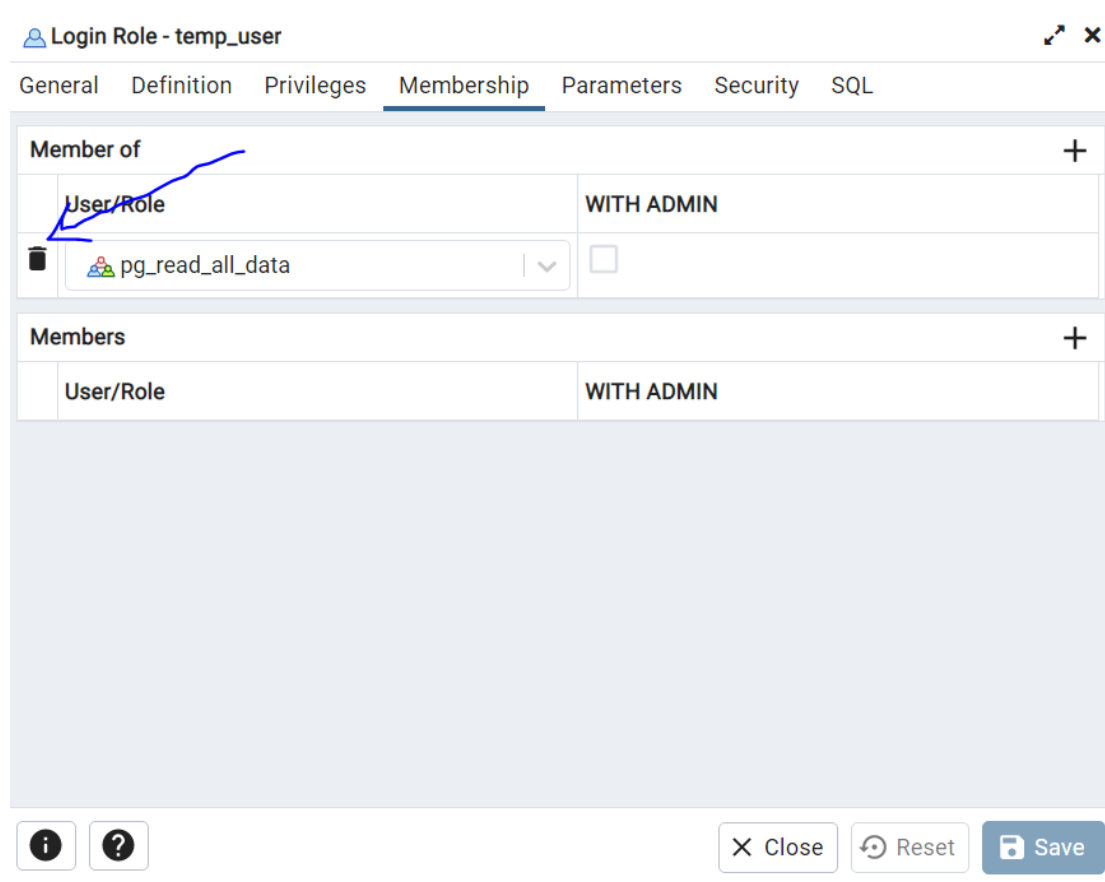



Рис. 10. Удаление роли

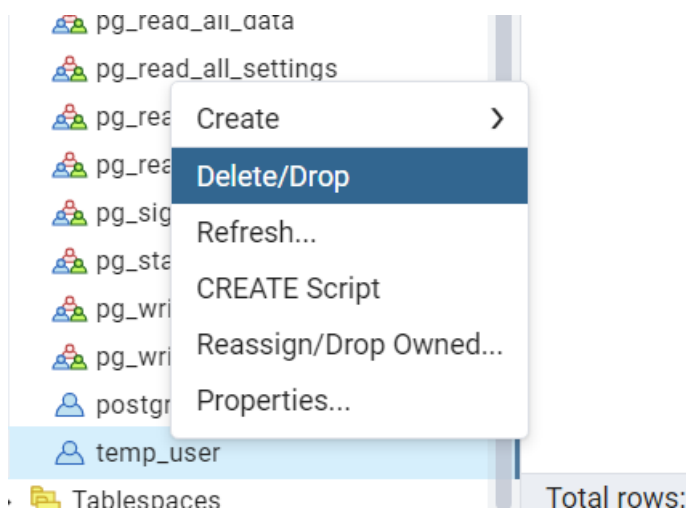


Рис. 11. Удаление пользователя

ВАЖНЫЙ МОМЕНТ! Удалить CrippledUser можно будет только после того, как у него будут забраны права:

```
REVOKE ("FirstColumn", "SecondColumn") ON public."TempTable" FROM "CrippledUser"
```

Самостоятельная работа

1. Определите всех пользователей, которые могут создавать базы данных (rolcreatedb).
2. Определите всех пользователей, которые могут создавать роли (rolcreatorole).
3. Создайте пользователя для входа с подключением к вашей базе данных, докажите правильность выполненных действий. Созданному пользователю присвойте права на создание и изменение баз данных, докажите правильность выполненных действий. Подключитесь к MS SQL Server, используя созданную учетную запись, и создайте еще одну учетную запись пользователя для входа, результат объясните.
4. Для созданного пользователя измените пароль.
5. Создайте пользователя *Admin* и присвойте ему роль, обладающую полным доступом к базе данных.
6. Создайте пользователя *User* и присвойте ему роль, обладающую доступом к базе данных только для чтения. (НЕ ЗАПИСЕЙ! ПРОСМОТР ТАБЛИЦ И ИХ КОЛОНОК!)
7. Пользователю *manager* присвойте роль, обладающую только возможностью просмотра содержимого вашей базы данных

Замечание. Для проверки правильности выполненных действий можно выполнить произвольный запрос к этой базе данных, например, отображающий содержимое таблицы *таблица1* (пример):
SELECT * FROM таблица1.

8. Пользователю *manager* запретите просмотр данных БД, присвоив необходимую роль. Как доказать правильность внесенных изменений?
9. В базе данных создайте пользователя на основе созданной ранее учетной записи для входа.
10. Для созданного ранее пользователя базы данных определите, членом какой роли он является и каково ее назначение. Имеет ли данный пользователь право выборки данных из *таблицы1* этой базы данных? Ответ обоснуйте и проверьте, выполнив извлечение данных командой SELECT * from таблица1 (пример).

11. В базе данных создайте роль *managers*. Для этой роли определите право выборки данных из таблицы *таблица1* базы данных. Присвойте роль *managers* созданному ранее пользователю. Имеет ли теперь этот пользователь право выборки данных? Проверьте сделанный вывод. К каким еще объектам базы данных имеет право доступа этот пользователь? Обоснуйте и проверьте вывод.

12. Создайте пользователя, имеющего доступ к вашей базе данных и принадлежащего роли *clerks*. Для этой роли определите возможность выборки данных из таблицы *таблица2* только для определенных полей, например, полей *Имя* и *Количество*. Для проверки правильности выполненных действий выполните команды:

- `SELECT * from Таблица1` – чтение данных из всех полей таблицы *Authors*;
- `SELECT Имя, Количество from Таблица1` – чтение данных таблицы *Таблица1* только из указанных полей.

13. Для роли *clerks* запрещена выборка данных из таблицы *Таблица1* базы данных. Пользователь *Andy* принадлежит пользовательской роли *clerks* и системной роли *pg_read_all_data*. Может ли этот пользователь получить данные из этой таблицы?