PostgresPro

Админ без доступа к данным

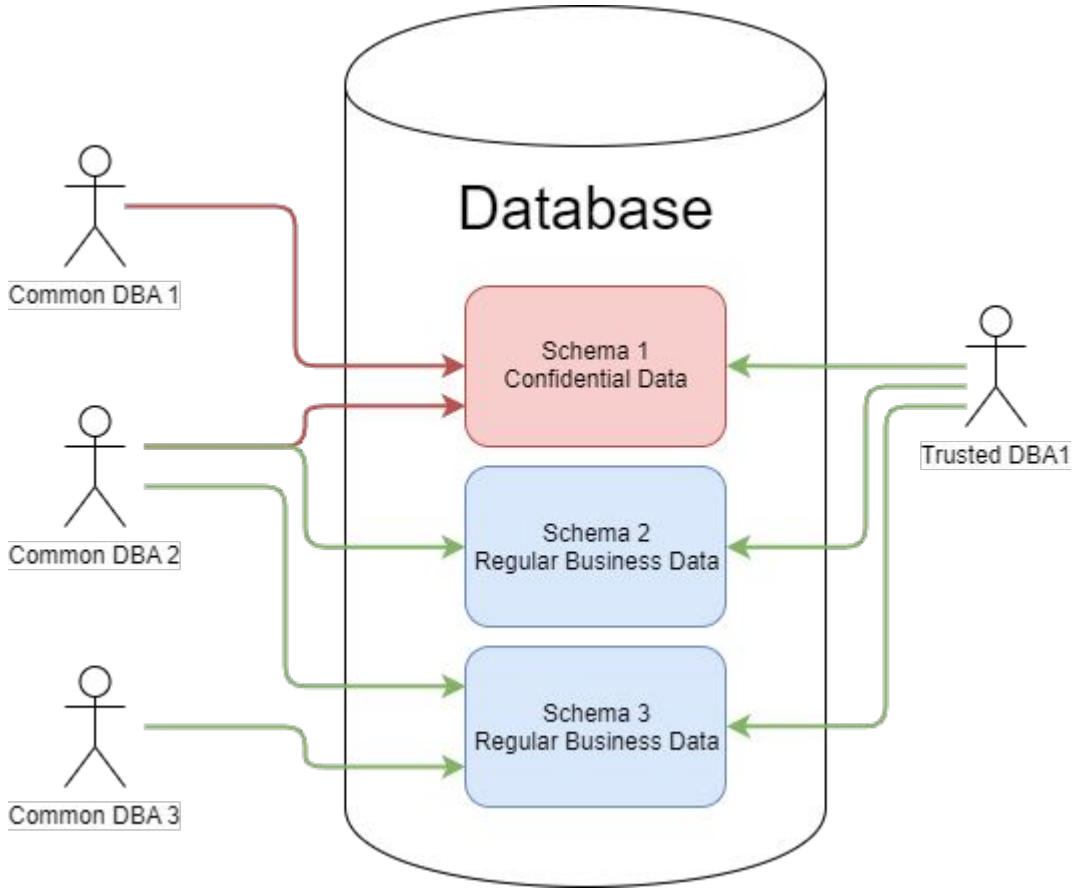
Ограничение доступа Администратора СУБД к данным

Василий Бернштейн

Postgres Professional

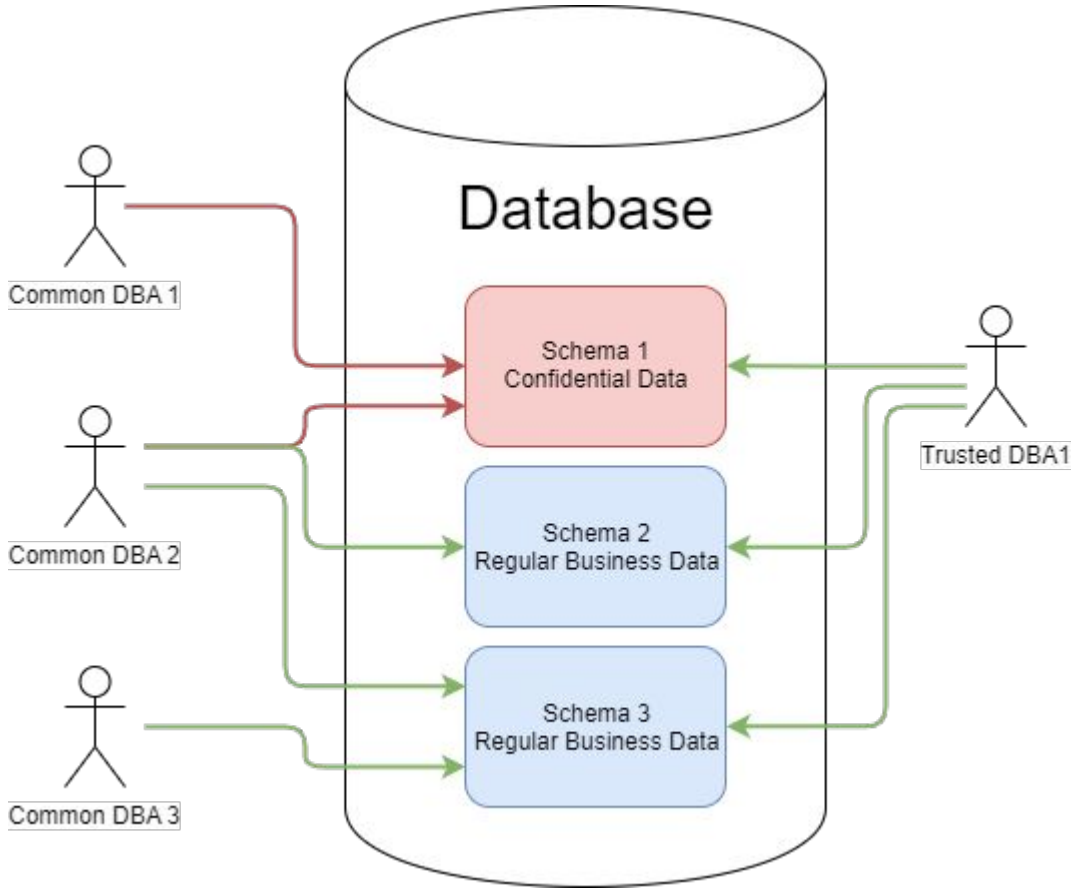
Проблема, которую надо решить

- Проблема возникает в больших организациях с «общими» администраторами баз данных
- Отдел DBA – в нём работают десятки, а то и сотни администраторов СУБД
- В таких организациях бывают ещё и «выделенные» администраторы, отвечающие именно за данные коммерческой тайны



Проблема, которую надо решить – пример

- Доступ к серверу имеют 20 человек DBA
- Они могут читать любые данные на сервере, чтобы решать возникающие на нём проблемы
- Есть таблица, в которой лежат персональные данные
- Любой из 20 администраторов может SELECT эти данные
- ИБ не может гарантировать, что никто из этих 20 DBA не украдёт и не «сольёт» эти данные
- А может просто по дурости запостить скриншот с куском данных в какой-нибудь форум



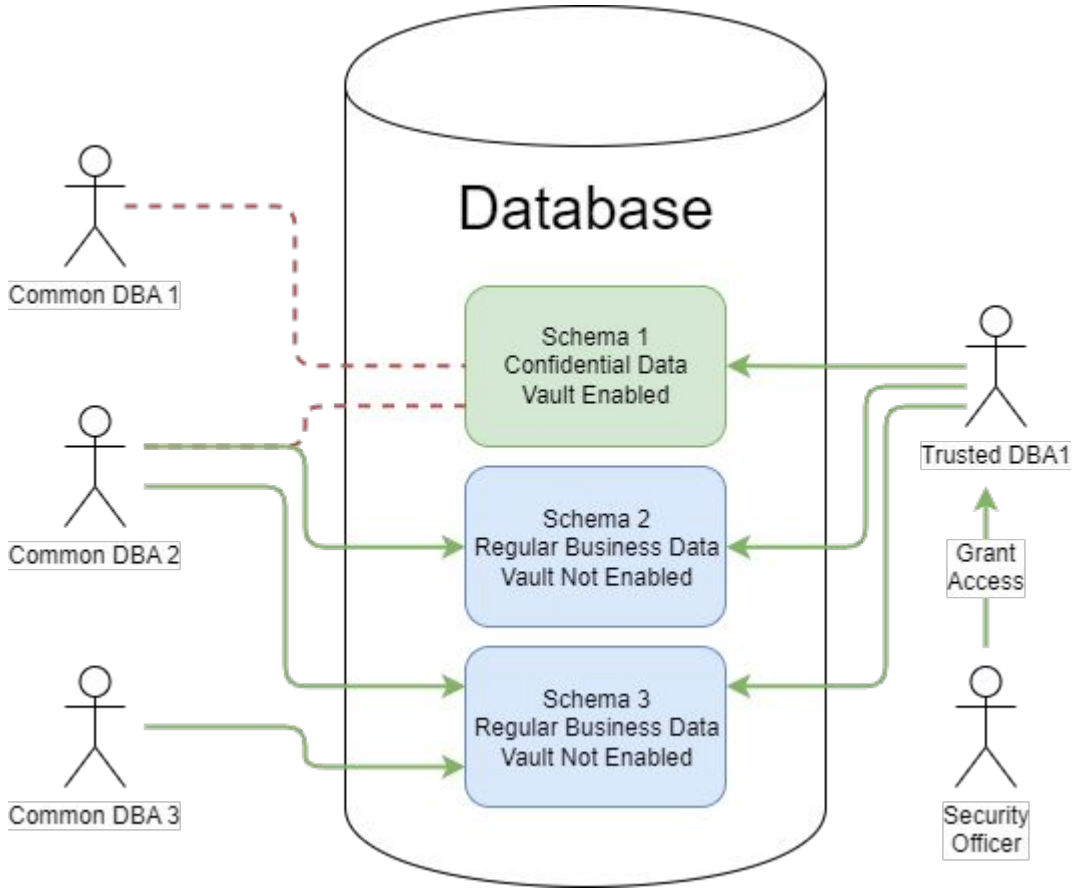
Постановка задачи

Две простые вещи:

- «Общие» администраторы СУБД не могут видеть и тем более менять чувствительные данные и данные коммерческой тайны
- «Доверенные» администраторы могут работать с чувствительными данными и коммерческой тайной

Что ещё?

- В больших организациях доступ к БД раздаёт отдел безопасности, а не Админы СУБД
- Нужно, чтобы доступ к чувствительным данным и коммерческой тайне давали тоже только они



Обзор решений других производителей

Oracle

- Secure Realm
- Есть две выделенные роли – DV_OWNER (владелец данных) и DV_ADMIN (администратор безопасности)
- Защищаемые объекты помещаются в Secure Realm
- Специальная сборка (подгружаемая библиотека), которая включает режим Secure Realm

PostgreSQL forks других производителей

- Ограничения доступа, которые распространяются и на superuser
- Дополнительная к ACL проверка доступа к данным
- Есть специальная системная роль, которая может управлять доступом к защищённым таблицам
- Ограничения устанавливаются на уровне таблиц

Недостатки подхода других PostgreSQL forks

PostgresPro

Очень сложно – для superuser

- Надо ограничить весь «очевидный» DML
- Надо ограничить все операции про ACL
- Надо ограничить все «опасные» системные функции

Ненадёжно

- Superuser может обойти механизм ограничений доступа
- Потому что на то он и superuser

И не быстро

- В дополнение к стандартному ACL надо проверять ещё и дополнительные политики и/или другие правила защиты

Например, только в aclchck.c нужно внести изменения в более чем 30 функций:

```
pg_class_aclmask_ext();
pg_database_aclmask();
pg_parameter_aclmask();
pg_parameter_acl_aclmask();
pg_proc_aclmask();
pg_language_aclmask();
pg_largeobject_aclmask_snapshot();
pg_namespace_aclmask();
pg_tablespace_aclmask();
pg_foreign_data_wrapper_aclmask();
pg_foreign_server_aclmask();
pg_type_aclmask();
pg_class_ownercheck();
pg_type_ownercheck();
pg_oper_ownercheck();
pg_proc_ownercheck();
pg_language_ownercheck();
pg_largeobject_ownercheck();
pg_namespace_ownercheck();
pg_tablespace_ownercheck();
pg_opclass_ownercheck();
pg_opfamily_ownercheck();
pg_ts_dict_ownercheck();
pg_ts_config_ownercheck();
pg_foreign_data_wrapper_ownercheck();
pg_foreign_server_ownercheck();
pg_event_trigger_ownercheck();
pg_database_ownercheck();
pg_collation_ownercheck();
pg_conversion_ownercheck();
pg_extension_ownercheck();
pg_publication_ownercheck();
pg_subscription_ownercheck();
pg_statistics_object_ownercheck();
```

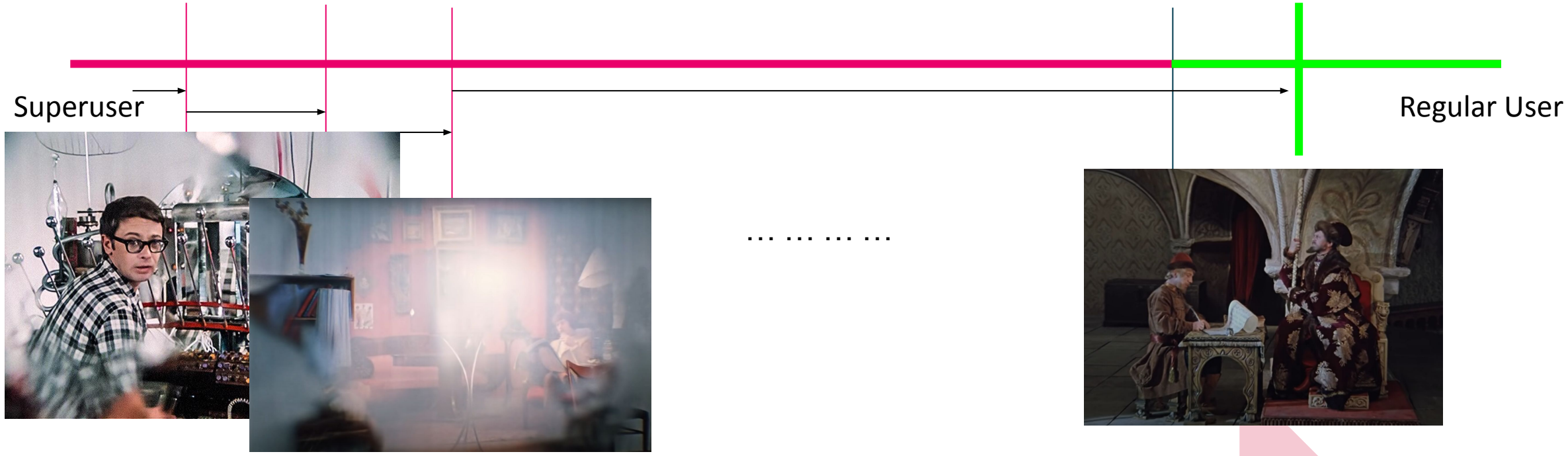
Исправление уязвимостей ограничений superuser

После исправления всех уязвимостей наш superuser будет мало отличаться от regular user

Он не сможет полноценно обслуживать систему во всех возможных ситуациях

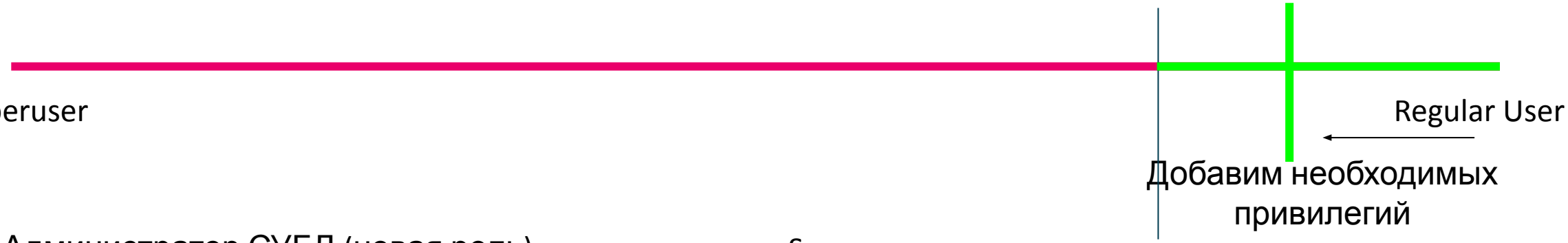
И нам понадобится какой-то новый super-superuser…

Исправление уязвимостей ограничений superuser



Цепочка исправлений приводит нас к принципиально другому состоянию, а не к модифицированному изначальному

Вместо ограничения привилегий – будем их добавлять



Администратор СУБД (новая роль)

- Технически – обычный пользователь
- Плюс необходимые для администрирования привилегии
- *Ежедневное администрирование СУБД без superuser становится новым стандартным подходом в vanilla*

Superuser

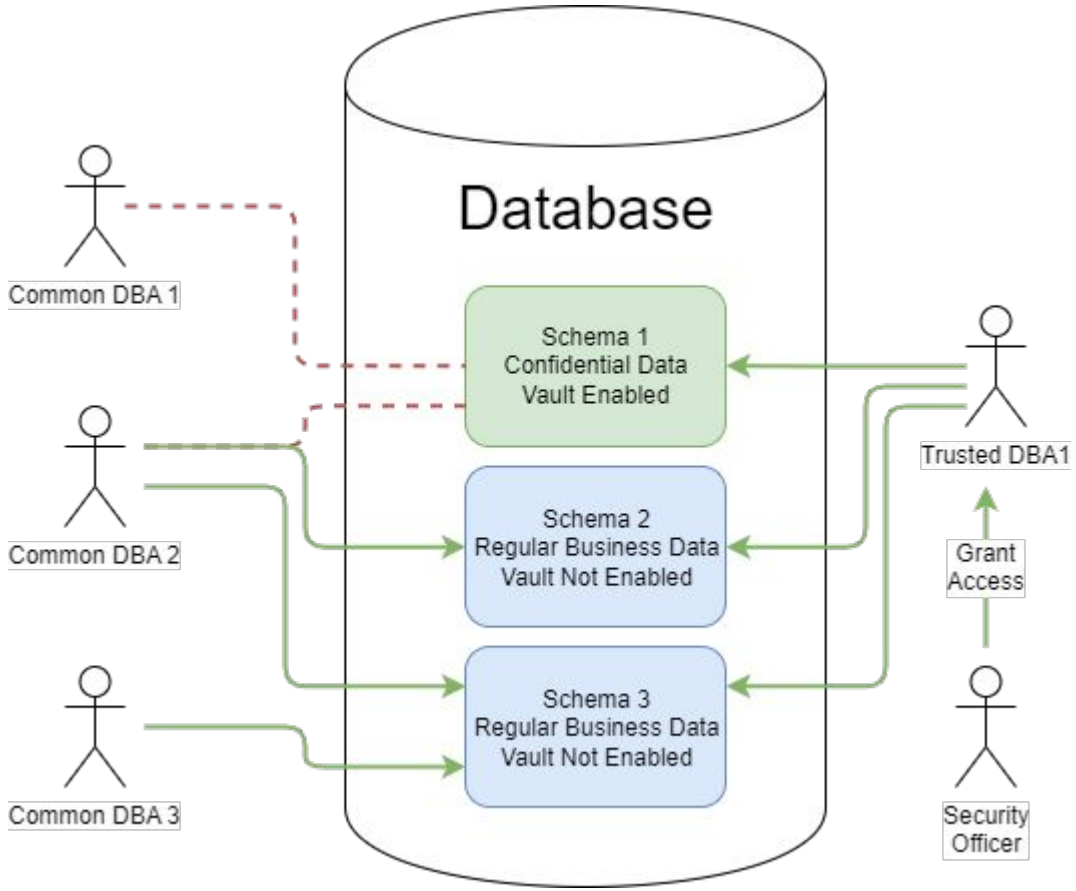
- Остаётся полноценным суперпользователем
- Не участвует в регулярных действиях с сервером
- Логин в базу заблокирован другими администраторами

Снятие ограничений superuser в «ограничительном» подходе равнозначно разрешению login в «разрешительном»

Механизм ограничения доступа - SCHEMA

Ограничение на уровне схемы

- Контейнер для конфиденциальных данных
- Один общий владелец
- Выделенный администратор безопасности



Защита через стандартный ACL – минимум изменений

Высокая надёжность
- ACL тестирует весь мир
- ограничения PostgreSQL forks – только их пользователи

Роль SECURITY OFFICER в Schema в явном виде
- Дополнительное поле pg_namespace
- Легко видна по стандартной команде \dn+

Изменения нужны
- Код выдачи прав на доступ к объектам
- В структуре каталога pg_namespace
- В синтаксисе ALTER SCHEMA
- pg_dump

```
postgres=> \c postgres dv_owner;
You are now connected to database "postgres" as user "dv_owner".
postgres=> grant select on vault_schema.vault_table to regular_user;
WARNING:  no privileges were granted for "vault_table"
GRANT
postgres=> \c postgres dv_admin;
You are now connected to database "postgres" as user "dv_admin".
postgres=> grant select on vault_schema.vault_table to regular_user;
GRANT
```

```
postgres=# CREATE SCHEMA vault_schema;
CREATE SCHEMA
postgres=# CREATE USER DV_OWNER;
CREATE ROLE
postgres=# CREATE USER DV_ADMIN;
CREATE ROLE
postgres=# ALTER SCHEMA vault_schema OWNER TO DV_OWNER;
ALTER SCHEMA
postgres=# ALTER SCHEMA vault_schema SECURITY OFFICER TO DV_ADMIN;
ALTER SCHEMA
postgres=# \dn+ vault_schema
                                 List of schemas
     Name     |  Owner   | Security officer | Access privileges | Description
--------------+----------+------------------+-------------------+-------------
 vault_schema | dv_owner | dv_admin         |                   |
(1 row)
```

Ролевая модель

Роль	Описание
Администратор безопасности системы	Администратор безопасности системы, на которой установлена СУБД. Это не администратор СУБД, а администратор инфраструктуры, на которой развёрнута СУБД. Не имеет доступа в СУБД.
PGPRO_DBMS_ADMIN	Администратор СУБД. Отвечает за: • управление сервером и репликацию • создание отдельных баз данных • не имеет доступа в защищённую зону
PGPRO_DB_X_ADMIN	Администратор БД "X". Отвечает за: • создание таблиц и других объектов в отдельной базе данных • создание пользователей БД • наделение пользователей правами доступа (за исключением объектов защищённой зоны) • не имеет доступа в защищённую зону
DV_OWNER	Владелец защищённой зоны базы данных: • DDL объектов защищённой зоны • Доступ к данным в защищённой зоне • Не может давать права доступа к объектам защищённой зоны
DV_SEC_OFFICER	Администратор безопасности защищённой зоны базы данных: • Только управляет правами доступа к объектам защищённой зоны

Ссылки

- Статья на Habr «Как в СУБД реализовать администратора без прав доступа к данным»
  https://habr.com/ru/companies/postgrespro/articles/788268/

- Схема vault (Postgres Pro Enterprise 16.2.1)
  https://postgrespro.ru/docs/enterprise/16/ddl-schemas#DDL-SCHEMAS-VAULT
- Расширение ALTER SCHEMA
  https://postgrespro.ru/docs/enterprise/16/sql-alterschema
- Дополнительные роли pg_create_tablespace, pg_manage_profiles
  https://postgrespro.ru/docs/enterprise/16/predefined-roles
- Новый параметр pg_dump “--privileges-only”
  https://postgrespro.ru/docs/enterprise/16/app-pgdump

- Статья Robert Haas “Surviving Without A Superuser - Coming to v16”
  https://www.enterprisedb.com/blog/surviving-without-Postgres-superuser

- Тред “MAINTAIN privilege -- what do we need to un-revert it?” для v17 (автор Jeff Davis)
  https://www.postgresql.org/message-id/d4ccaf3658cb3c281ec88c851a09733cd9482f22.camel@j-davis.com

Отвечу на ваши вопросы

PostgresPro

@VASILY_BERNSTEIN

PostgresPro

Спасибо за внимание!

postgrespro.ru

PostgresPro

Q&A