

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ «JATOVA»

Руководство по настройке. Часть 2.
Контроль субъектов доступа.
Компонент «Jatoba data vault»

643.72410666.00067-07 98 01-02

Листов 29

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

В документе приведены сведения, необходимые для установки и эксплуатации компонента «Jatoba data vault» (далее – Компонент JDV). Настоящее руководство предназначено для администратора СУБД «Jatoba».

Администратор СУБД «Jatoba» должен иметь навыки по работе с системами управления базами данных (СУБД) PostgreSQL или защищенной СУБД «Jatoba» (ООО «Газинформсервис»).



Все примеры в данном документе приведены для СУБД «Jatoba» версии ядра 4.x, для других версий все шаги выполняются аналогично, разница состоит в именах директорий.

Например, СУБД «Jatoba» версии 5.x по умолчанию устанавливается в директорию ОС Linux – «/usr/jatoba-5/bin».

Для СУБД «Jatoba» версии ядра 4 используется версия компонента — 1.5

Для СУБД «Jatoba» версии ядра 5/6 используется версия компонента — 1.6



Важная информация

Для сертифицированной версии СУБД «Jatoba» поддерживается работа только на ОС, указанных в формуляре на поставку!

Степени важности примечаний, применяемые в документе:



Важная информация – указания, требующие особого внимания



Дополнительная информация – указания, позволяющие упростить работу с изделием

СОДЕРЖАНИЕ

1. Назначение компонента.....	4
1.1. Функциональные возможности	4
2. Установка и настройка компонента	7
2.1. Установка компонента в ОС GNU/Linux.....	7
2.2. Установка расширения JDV.....	8
2.2.1. Особенности совместного использования компонентов JDV и securityprofile	8
2.2.2. Установка расширения JDV отдельно	9
2.3. Использование компонента	11
2.3.1. Роль «dv_owner».....	13
2.3.2. Групповая роль «dv_secanalyst».....	13
2.3.3. Роль «dv_acctmgr»	14
2.3.4. Групповая роль «dv_group».....	15
3. Проверка работоспособности компонента.....	16
4. Описание операций.....	20
4.1. Управление списком защищаемых таблиц.....	20
4.2. Управление списком защищенных ролей	20
4.3. Управление функциями мониторинга	21
4.3.1. Определение событий безопасности подлежащих регистрации компонентом JDV	21
4.3.2. Управление регистрацией событий безопасности (логированием) компонента JDV	22
4.3.3. Управление местом хранения событий безопасности и политикой хранения	24
5. Временное отключение компонента	26
6. Удаление компонента	27
Перечень сокращений.....	28

1. НАЗНАЧЕНИЕ КОМПОНЕНТА

Компонент JDV предназначен для ограничения доступа пользователей СУБД к защищаемым объектам баз данных (БД).

1.1. Функциональные возможности

Компонент JDV позволяет создать список защищаемых таблиц БД, работать с каждой из которых могут только:

- владелец таблицы;
- пользователь, который имеет доступ к таблице и при этом имеет специальное разрешение (команда `jdv_set_perm`).

Пользователи с полными правами в СУБД (далее – суперпользователи) не имеют доступа к защищенным таблицам, если они не относятся к данным категориям.

Для суперпользователей по отношению к защищаемым объектам БД недоступны команды:

- SELECT;
- INSERT;
- UPDATE;
- DELETE;
- DROP;
- TRUNCATE.

Суперпользователям недоступны команды:

- CREATE ROLE, DROP ROLE, ALTER ROLE;
- CREATE EXTENSION;
- CREATE TRIGGER;
- DROP EXTENSION `jdv`;
- LOAD.

Суперпользователям частично недоступны команды:

- DROP OWNED;
- GRANT;
- REASSIGN OWNED;
- SET ROLE;
- SET SESSION AUTHORIZATION.

Данные команды недоступны, если они применяются по отношению:

- к ролям, владеющим защищаемыми объектами;
- к ролям, имеющим специальные разрешения;
- к служебным объектам расширения.

Суперпользователю недоступны для изменения и удаления само расширение и объекты расширения:

- схема (jdv);
- роли (dv_owner, dv_secanalyst, dv_acctmgr, dv_group);
- таблицы и функции расширения.

Суперпользователю недоступны команды INSERT, UPDATE, DELETE по отношению к системным каталогам:

- pg_attribute;
- pg_authid;
- pg_auth_members;
- pg_constraint;
- pg_db_role_setting;
- pg_enum;
- pg_extension;
- pg_index;
- pg_init_privs;
- pg_namespace;

- pg_proc;
- pg_class;
- pg_type.



Компонент JDV добавляет ограничения при работе с объектами, но не предоставляет дополнительных средств для обхода стандартной системы проверки доступа к объектам

Для получения доступа к защищаемой таблице пользователь должен иметь:

- 1) доступ от СУБД «Jatoba»;
- 2) разрешение от компонента JDV.

Суперпользователь должен иметь разрешение от JDV.

2. УСТАНОВКА И НАСТРОЙКА КОМПОНЕНТА

Компонент JDV устанавливается на ЭВМ, на которой установлена расширяемая СУБД.

Все команды при установке и при работе с компонентом JDV выполняются в консоли работы с СУБД (встроенная в PostgreSQL и СУБД «Jatoba» утилита psql).

2.1. Установка компонента в ОС GNU/Linux

Компонент устанавливается в составе СУБД «Jatoba». Его возможно установить при первичной установке, либо доустановить.

Установку компонента возможно провести двумя способами:

- 1) установка из локального репозитория (CDROM) – производится из файлов, записанных на компакт-диск или скопированных с него;
- 2) установка непосредственно из deb/rpm-файлов – производится опционально, по усмотрению пользователя.

Компонент выполнен в виде отдельного deb или rpm-пакета. Установка компонента осуществляется средствами пакетного менеджера ОС. Для разных типов пакетных менеджеров команда установки немного отличается. Ниже приведены основные типы:

– для систем на основе пакетного менеджера APT (к таким системам относятся все ОС семейства Debian, использующие deb-пакеты) команда установки следующая:

```
apt-get install jatoba4-jdv
```

– для систем на основе пакетных менеджеров YUM/DNF (к таким системам относятся все ОС семейства RedHat и вышедшие из нее, использующие rpm-пакеты) команда установки следующая:

```
yum install jatoba4-jdv
```

Отдельного уточнения требуют операционные системы ALT Linux и openSUSE.

– ALT Linux использует пакетный менеджер APT, но распространяется в виде rpm-пакетов и для нее команда установки выглядит аналогично Debian:

```
apt-get install jatoba4-jdv
```

Установка компонента в составе других версий СУБД «Jatoba» осуществляется аналогично. Отличие будет только в номере версии СУБД, в составе которой он распространяется. Например, jatoba4-jdv т.п.

Удаление модуля также осуществляется средствами пакетного менеджера ОС. Вместо команды install нужно использовать соответствующую данному пакетному менеджеру команду удаления (remove, purge, erase и т.п.).

Для получения детальной информации по пакетному менеджеру рекомендуется обратиться к документации по ОС.

2.2. Установка расширения JDV

2.2.1. Особенности совместного использования компонентов JDV и securityprofile

Компонент JDV совместим с компонентом SecurityProfile версии 1.2 и старше.

При использовании компонента парольных политик SecurityProfile, сначала необходимо установить расширение SecurityProfile, затем расширение jdv.

В конфигурационном файле в «postgresql.conf» установить следующий порядок загрузки:

```
shared_preload_libraries = 'jdv, securityprofile'
```

После чего выполнить следующие действия:

1) Указать в конфигурационном файле «pg_hba.conf» метод «md5» для ipv4 подключений.



Начиная с СУБД «Jatoba» версии 18 для компонента SecurityProfile необходимо применять метод аутентификации «scram-sha-256».

2) Выполнить перезагрузку СУБД.

3) Авторизоваться в СУБД от имени и с правами суперпользователя.

4) Создать расширение securityprofile SQL-командой:

```
CREATE EXTENSION securityprofile;
```

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

- 5) Выполнить создание профиля dv_profile парольной политики компонента SecurityProfile в соответствии с указаниями из п.п. 6.1.6. Взаимодействие с компонентом JDV из документа «Руководство администратора» 643.72410666.00067-07 95 01;

- 6) Создать расширение jdv SQL-командой:

```
CREATE EXTENSION IF NOT EXISTS jdv;
```

- 7) Функцию инициализации securityprofile SQL-командой:

```
SELECT securityprofile.synchronize();
```

- 8) Установить пароль для суперпользователя, подходящий под установленную парольную политику по умолчанию.
- 9) Авторизоваться под служебными пользователями JDV – dv_acctmgr или dv_owner.

2.2.2. Установка расширения JDV отдельно

Для установки компонента выполняется следующая последовательности действий:

- 1) В разделе «Shared Library Preloading» конфигурационного файла postgresql.conf внести изменения:

```
shared_preload_libraries = 'jdv'
```

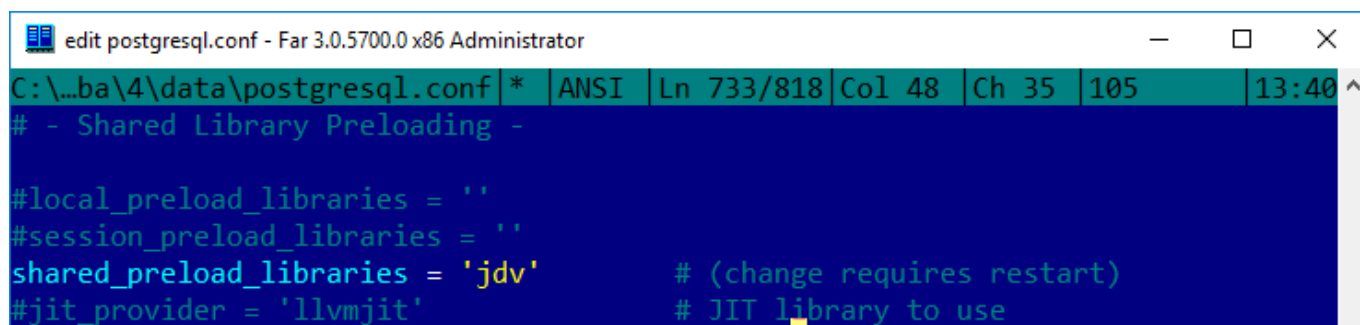


Рисунок 2.1 – Окно редактирования postgresql.conf

Затем для применения параметров перезагрузить СУБД.

- 2) Загрузить расширение, выполнив команду:

```
CREATE EXTENSION IF NOT EXISTS jdv;
```

```

Администратор: Командная строка - psql -h localhost -d postgres -U postgres

c:\Program Files\GIS\Jatoba4\bin>psql -h localhost -d postgres -U postgres
Пароль пользователя postgres:
psql (14.5)
Введите "help", чтобы получить справку.

postgres=# CREATE EXTENSION IF NOT EXISTS jdv;
CREATE EXTENSION
postgres=#
  
```

Рисунок 2.2 – Создание расширения



Загрузить расширение будет невозможно, если не выполнен п. 2.

В результате установки расширения в СУБД будут созданы:

– схема jdv;

```

Администратор: Командная строка - psql -h localhost -d postgres -U postgres

postgres=# \dx
                                Список установленных расширений
  Имя  | Версия | Схема | Описание
-----+-----+-----+-----
ja_sync_ldap | 1.1 | public | supports account sync with AD/LDAP
jdv    | 1.4 | public | The jdv module provides additional protection for the tables
plpgsql | 1.0 | pg_catalog | PL/pgSQL procedural language
(3 строки)

postgres=#
  
```

– роли:

- dv_owner (временный пароль: P@ssword1);
- dv_acctmgr (временный пароль: P@ssword2);
- dv_secanalyst (nologin);

```

Администратор: Командная строка - psql -h localhost -d postgres -U postgres

postgres=# \du
                                Список ролей
  Имя роли  | Атрибуты | Член ролей
-----+-----+-----
ad_users   | Вход запрещён | {}
admin_bd   | Не наследуется, Создаёт роли | {}
dv_acctmgr | Создаёт роли | {dv_group}
dv_group   | Не наследуется, Вход запрещён | {}
dv_owner   | | {dv_group,dv_secanalyst}
dv_secanalyst | Не наследуется, Вход запрещён | {dv_group}
jds        | Не наследуется | {}
postgres   | Суперпользователь, Создаёт роли, Создаёт БД, Репликация, Пропускать RLS | {}
  
```

– служебные таблицы:

- jdv_log;
- jdv_log_meta;
- jdv_log_rules;

- jdv_settings;
- jdv_table;

Администратор: Командная строка - psql -h localhost -d postgres -U postgres

```
postgres=# SELECT * FROM pg_catalog.pg_tables WHERE schemaname='jdv';
```

schemaname	tablename	tableowner	tablespace	hasindexes	hasrules	hastriggers	rowsecurity
jdv	jdv_table	dv_owner		t	f	f	f
jdv	jdv_log_rules	dv_owner		t	f	f	f
jdv	jdv_log	dv_owner		t	f	f	f
jdv	jdv_log_meta	dv_owner		f	f	f	f
jdv	jdv_settings	dv_owner		t	f	f	f

(5 строк)

Рисунок 2.3 – Служебные таблицы

- функции расширения:
- в схеме jdv.

Администратор: Командная строка - psql -h localhost -d postgres -U postgres

```
postgres=# \df jdv.*
```

Схема	Имя	Тип данных результата	Типы данных аргументов	Тип
jdv	jdv_activate	boolean		функ.
jdv	jdv_add_role	boolean	role_name name	функ.
jdv	jdv_add_table	boolean	table_name name	функ.
jdv	jdv_create_role	void		функ.
jdv	jdv_deactivate	boolean		функ.
jdv	jdv_log_dest	boolean	i text	функ.
jdv	jdv_log_exclude_object	boolean	VARIADIC object_names name[]	функ.
jdv	jdv_log_exclude_role	boolean	VARIADIC role_names name[]	функ.
jdv	jdv_log_exclude_schema	boolean	VARIADIC schema_names name[]	функ.
jdv	jdv_log_flush	boolean	k jdv_log_option, i integer DEFAULT 0	функ.
jdv	jdv_log_include_object	boolean	VARIADIC object_names name[]	функ.
jdv	jdv_log_include_role	boolean	VARIADIC role_names name[]	функ.
jdv	jdv_log_include_schema	boolean	VARIADIC schema_names name[]	функ.
jdv	jdv_remove_role	boolean	role_name name	функ.
jdv	jdv_remove_table	boolean	table_name name	функ.
jdv	jdv_reset_perm	boolean	table_name name, role_name name	функ.
jdv	jdv_set_perm	boolean	table_name name, role_name name	функ.

(17 строк)

postgres=#

Рисунок 2.4 – Функции расширения в схеме JDV



В случае последующей установки компонента SecurityProfile необходимо создать профиль dv_profile парольной политики в соответствии с указаниями из п.п. 6.1.6. Взаимодействие с компонентом JDV из документа «Руководство администратора» 643.72410666.00067-07 95 01

2.3. Использование компонента

Типовое соотношение структуры и атрибутов ролей в СУБД отображено на рисунке 2.5.

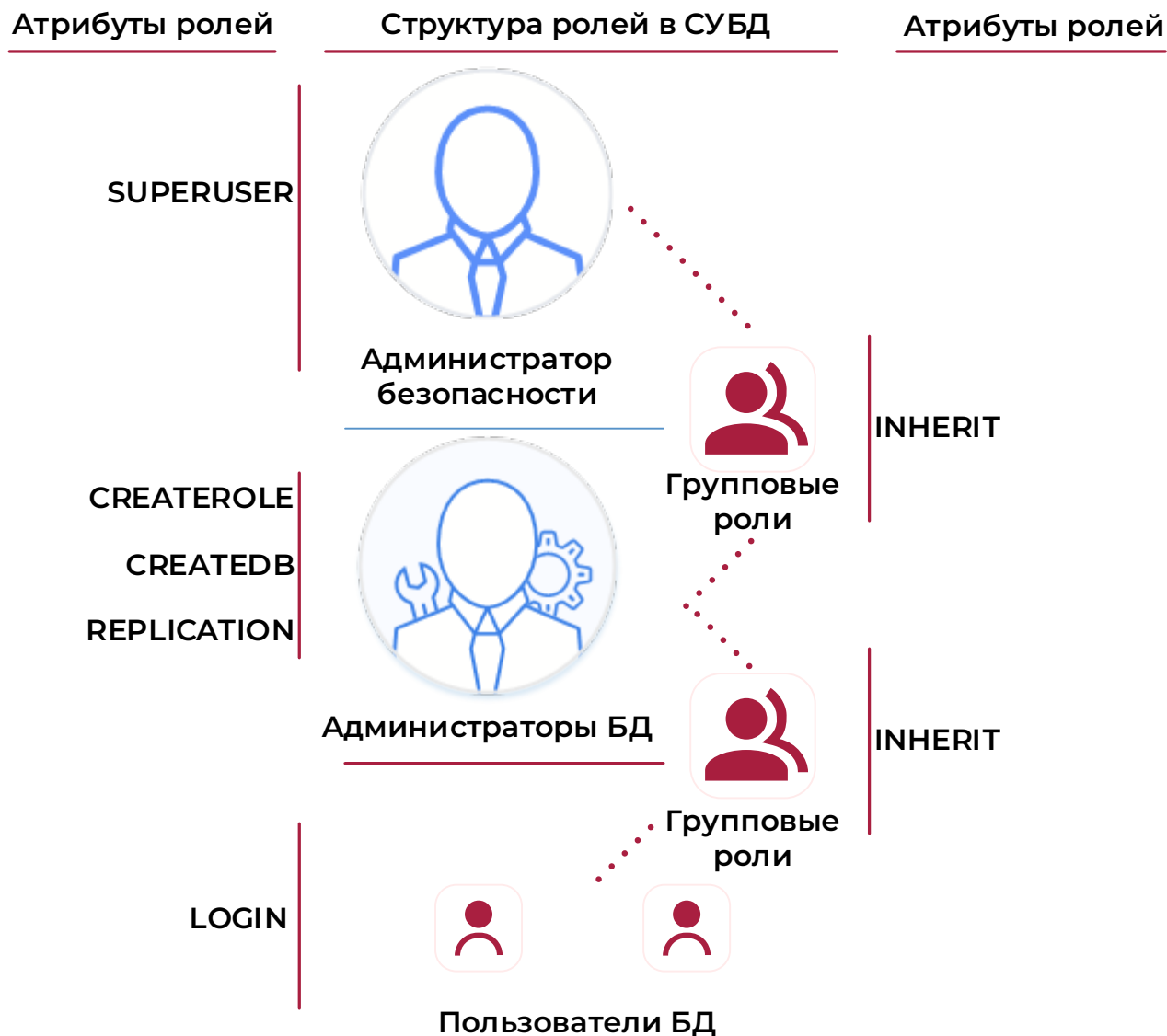


Рисунок 2.5 – Схема структуры атрибутов ролей

При активации компонентов JDV и SecurityProfile структура ролей управления СУБД изменяется. Компонент JDV предназначен для ограничения доступа пользователей СУБД к защищаемым объектам баз данных. Компонент SecurityProfile предназначен для реализации парольной политики.

Условная группа Администраторов баз данных делится на дополнительные роли с переходом функциональных возможностей по:

- администрированию защищаемых таблиц и пользователей;
- мониторинга ролей, объектов и схем;
- администрированию пользователей.

Ролевая модель состоит из:

- администратора баз данных (Database administrator);
- администратора безопасности (Security administrator);
- аудитора (Auditor);
- администратора пользователей (User administrator);
- защищаемых пользователей и пользователей (Protected users & users).

2.3.1. Роль «dv_owner»

Для администрирования защищаемых таблиц и пользователей администратор безопасности (Security administrator) имеет роль «dv_owner», при этом не имеет функциональных возможностей по администрированию пользователей и используется только для работы с функциями расширения.

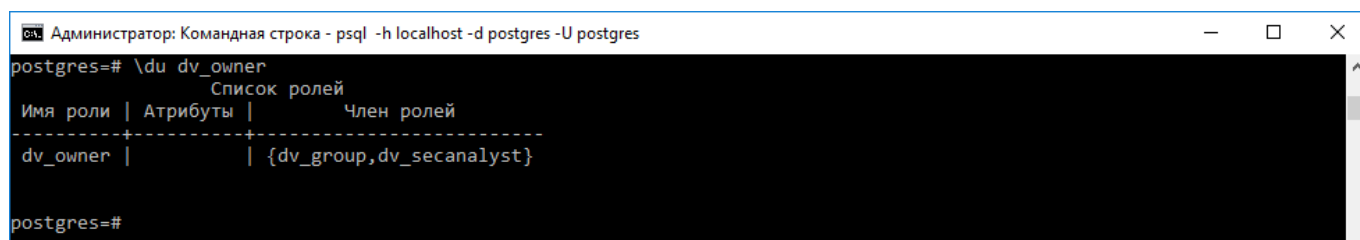


Рисунок 2.6 – Свойства роли «dv_owner»

При использовании компонента JDV совместно с

2.3.2. Групповая роль «dv_secanalyst»

Аудитор (Auditor) отнесен к групповой роли «dv_secanalyst», имеет функциональные возможности только просмотра событий безопасности, генерируемых компонентом JDV, таких как jdv_table, jdv_log_rules и jdv_log.

Роль «dv_secanalyst» не используется для обработки информации с СУБД.

Для работы с ней dv_owner должен предварительно включить существующую роль в члены этой группы (например, роль администратора безопасности, но не суперпользователя):

```
GRANT dv_secanalyst TO role AIB
```

```
Администратор: Командная строка - psql -h localhost -d postgres -U postgres
postgres=# \du dv_secanalyst
                Список ролей
Имя роли | Атрибуты | Член ролей
-----+-----+-----
dv_secanalyst | Не наследуется, Вход запрещён | {dv_group}
postgres=#
```

Рисунок 2.7 – Свойства групповой роли «dv_secanalyst»

2.3.3. Роль «dv_acctmgr»

Администратор пользователей (User administrator) отнесен к роли «dv_acctmgr» и имеет эксклюзивную функциональную возможность по администрированию пользователей и ряд функциональных особенностей:

- Роль dv_acctmgr эксклюзивно получает права на CREATE / ALTER / DROP / \password [username]' для всех ROLE / USER.
- Роль dv_acctmgr не может создавать/удалять суперпользователей и изменять их параметры.
- Роль dv_acctmgr не может изменить пароль для роли dv_owner и ее членов, а также суперпользователей. Все пользователи могут изменить свой пароль самостоятельно.



Все роли, которые имеют атрибут CREATEROLE (имеющиеся или вновь созданные), могут воспользоваться этой опцией только если являются членами группы dv_acctmgr.

Только dv_acctmgr может включить роль в dv_acctmgr (кроме суперпользователей и членов их групп):

```
GRANT dv_acctmgr TO role
```



Ограничение версии: dv_acctmgr не может делать ALTER RENAME для защищенных ролей.

```
Администратор: Командная строка - psql -h localhost -d postgres -U postgres
postgres=# \du dv_acctmgr
                Список ролей
Имя роли | Атрибуты | Член ролей
-----+-----+-----
dv_acctmgr | Создаёт роли | {dv_group}
postgres=#
```

Рисунок 2.8 – Свойства роли «dv_acctmgr»

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

2.3.4. Групповая роль «dv_group»

Защищаемые пользователи (Protected users), являющиеся владельцами таблиц(ы) и (или) имеющие специальное разрешение на доступ к защищаемому объекту, относятся к групповой роли «dv_group». Остальные пользователи (users) могут относиться к любым другим групповым ролям.

```

Выбрать Администратор: Командная строка - psql -h localhost -d postgres -U postgres
postgres=# \du dv_group
                Список ролей
Имя роли | Атрибуты | Член ролей
-----+-----+-----
dv_group | Не наследуется, Вход запрещён | {}
postgres=#

```

Рисунок 2.9 – Свойство групповой роли «dv_group»

Ролевая модель при функционировании компонентов JDV и SecurityProfile представлена на рисунке 2.10.

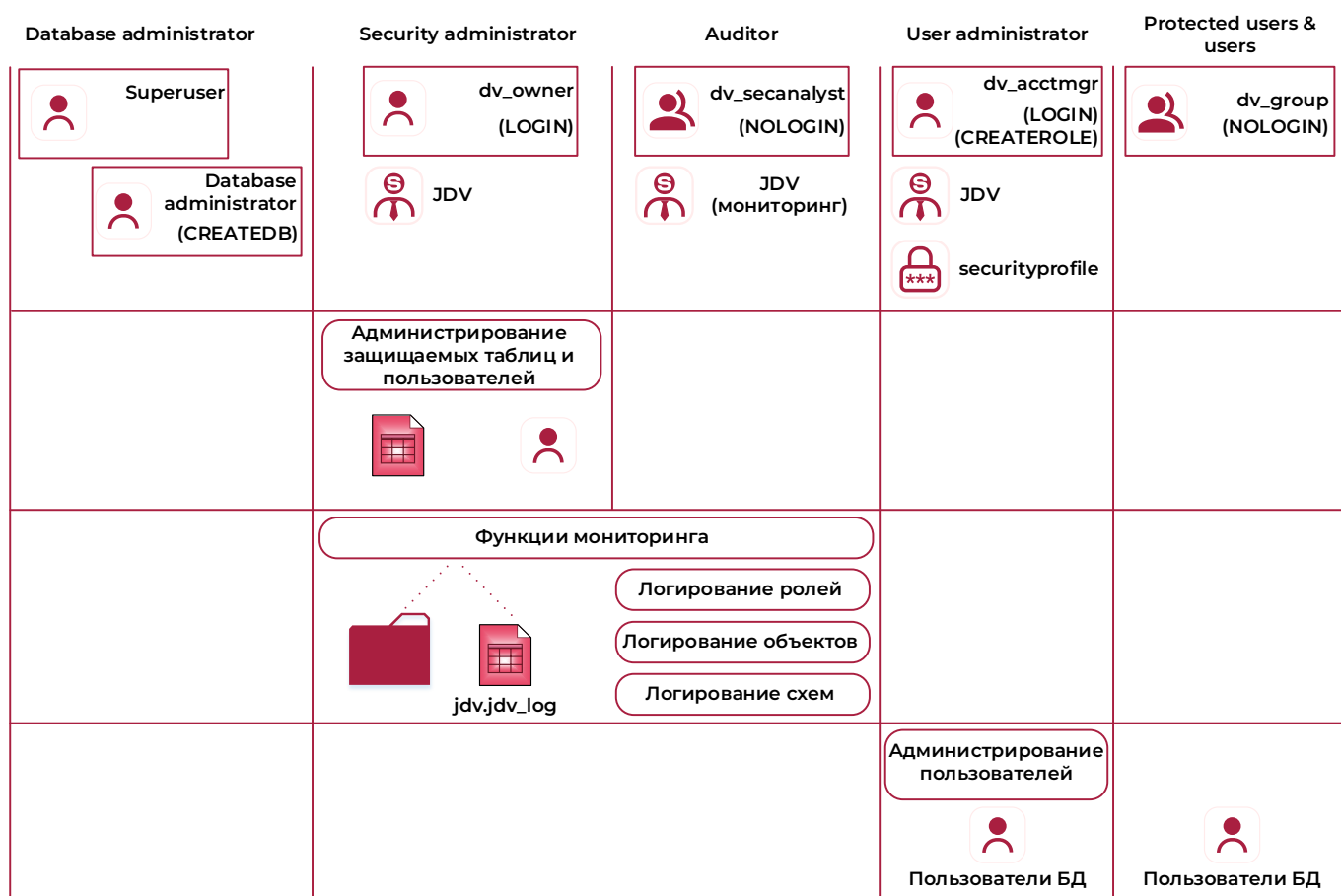


Рисунок 2.10 – Распределение функциональных возможностей

3. ПРОВЕРКА РАБОТОСПОСОБНОСТИ КОМПОНЕНТА

Для проверки корректности работы компонента JDV необходимо выполнить следующие действия:

1) Пользователь «dv_acctmgr» создает роли «role_x», «role_y» и «roleAIB», выполнив следующие команды:

```
CREATE ROLE role_x LOGIN PASSWORD 'P@ssword3';
```

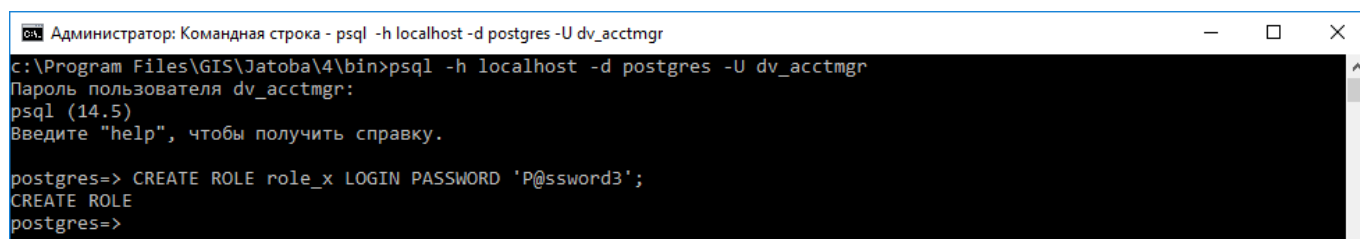


Рисунок 3.1 – Окно создания пользователя «role_x»

```
CREATE ROLE role_y LOGIN PASSWORD 'P@ssword4';
```

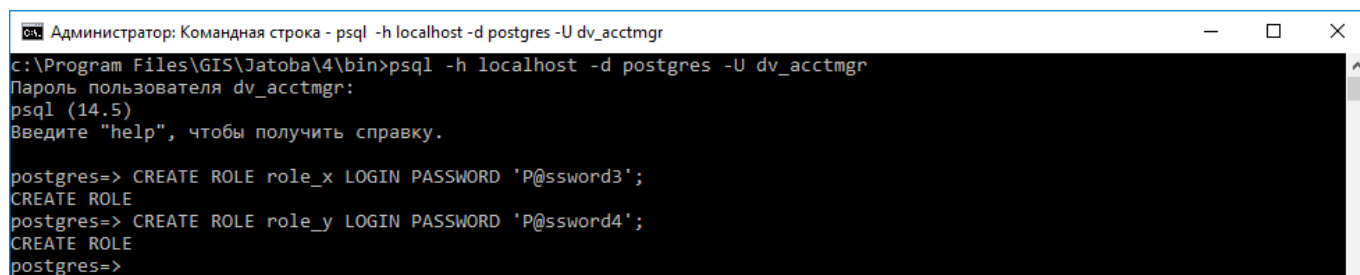


Рисунок 3.2 – Окно создания пользователя «role_y»

```
CREATE ROLE roleAIB NOSUPERUSER NOCREATEDB NOCREATEROLE  
NOINHERIT LOGIN PASSWORD 'P@ssword5';
```

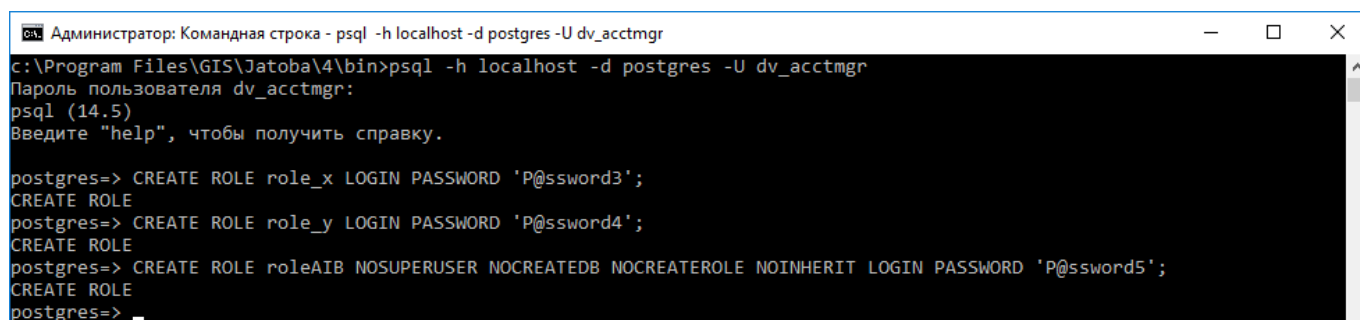
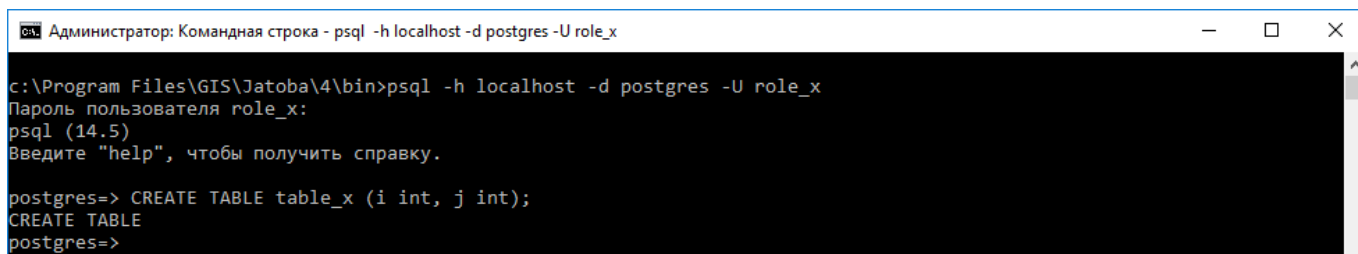


Рисунок 3.3 – Окно создания роли «roleAIB»

2) Пользователь «role_x» создает тестовую таблицу и устанавливает доступ к ней для пользователя «role_y»:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------


```
CREATE TABLE table_x (i int, j int);
```

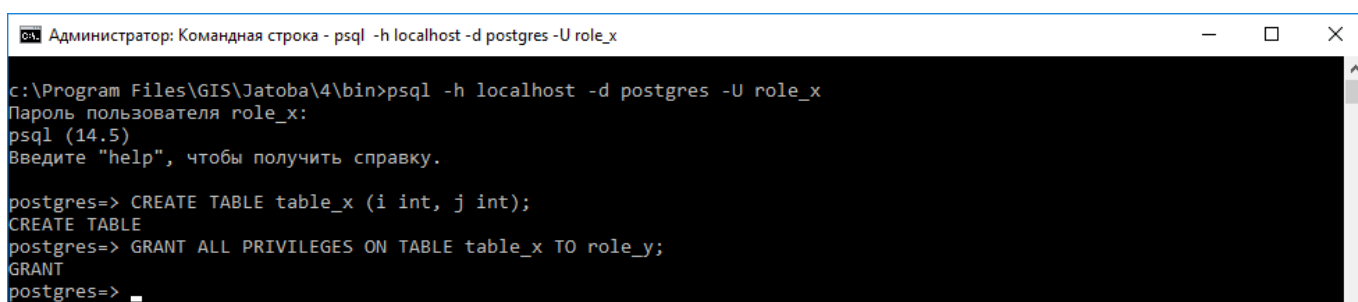


```
Администратор: Командная строка - psql -h localhost -d postgres -U role_x
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U role_x
Пароль пользователя role_x:
psql (14.5)
Введите "help", чтобы получить справку.

postgres=> CREATE TABLE table_x (i int, j int);
CREATE TABLE
postgres=>
```

Рисунок 3.4 – Окно создания таблицы table_x

```
GRANT ALL PRIVILEGES ON TABLE table_x TO role_y;
```



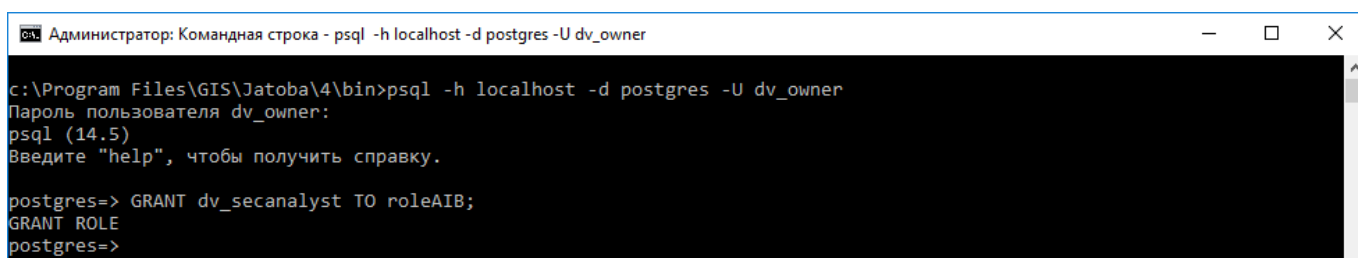
```
Администратор: Командная строка - psql -h localhost -d postgres -U role_x
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U role_x
Пароль пользователя role_x:
psql (14.5)
Введите "help", чтобы получить справку.

postgres=> CREATE TABLE table_x (i int, j int);
CREATE TABLE
postgres=> GRANT ALL PRIVILEGES ON TABLE table_x TO role_y;
GRANT
postgres=> _
```

Рисунок 3.5 – Окно предоставления привилегий пользователю «role_y»

3) Пользователь «dv_owner» включает в состав групповой роли «dv_secanalyst» роль-наблюдателя «roleAIB», выполнив команду:

```
GRANT dv_secanalyst TO roleAIB;
```



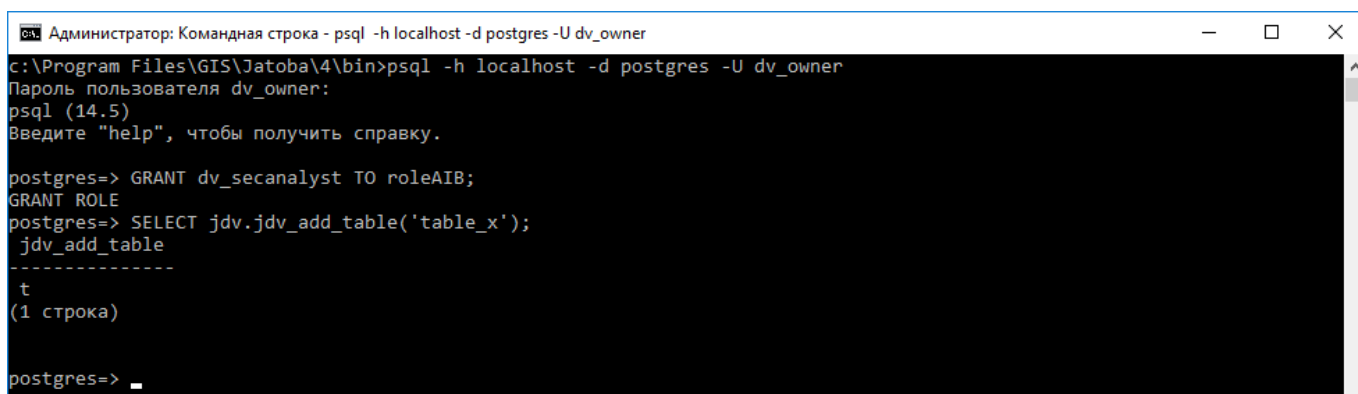
```
Администратор: Командная строка - psql -h localhost -d postgres -U dv_owner
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U dv_owner
Пароль пользователя dv_owner:
psql (14.5)
Введите "help", чтобы получить справку.

postgres=> GRANT dv_secanalyst TO roleAIB;
GRANT ROLE
postgres=>
```

Рисунок 3.6 – Окно включения роли «roleAIB» в состав групповой роли «dv_secanalyst»

4) Пользователь «dv_owner» включает таблицу «table_x» в список защищаемых объектов, выполнив команду:

```
SELECT jdv.jdv_add_table('table_x');
```



```
Администратор: Командная строка - psql -h localhost -d postgres -U dv_owner
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U dv_owner
Пароль пользователя dv_owner:
psql (14.5)
Введите "help", чтобы получить справку.

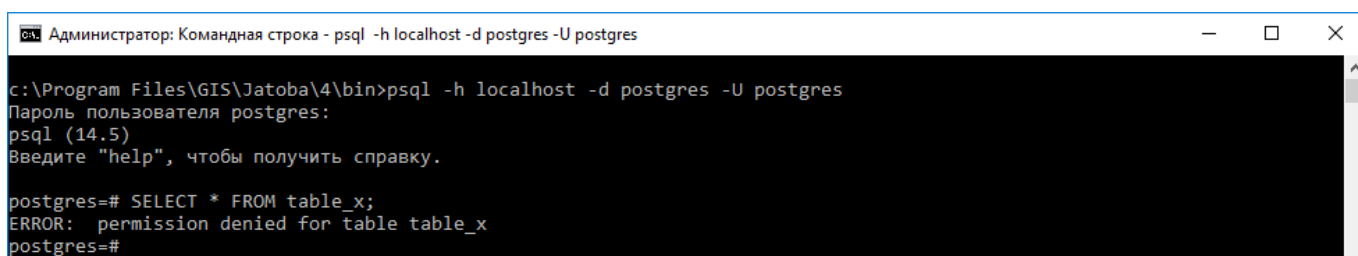
postgres=> GRANT dv_secanalyst TO roleAIB;
GRANT ROLE
postgres=> SELECT jdv.jdv_add_table('table_x');
 jdv_add_table
-----
 t
(1 строка)

postgres=> _
```

Рисунок 3.7 – Окно включения таблицы «table_x» в список защищаемых объектов

5) Пользователь «SUPERUSER» при попытке доступа к защищаемой таблице «table_x» получает отказ на выполнение операции:

```
SELECT * FROM table_x;
```



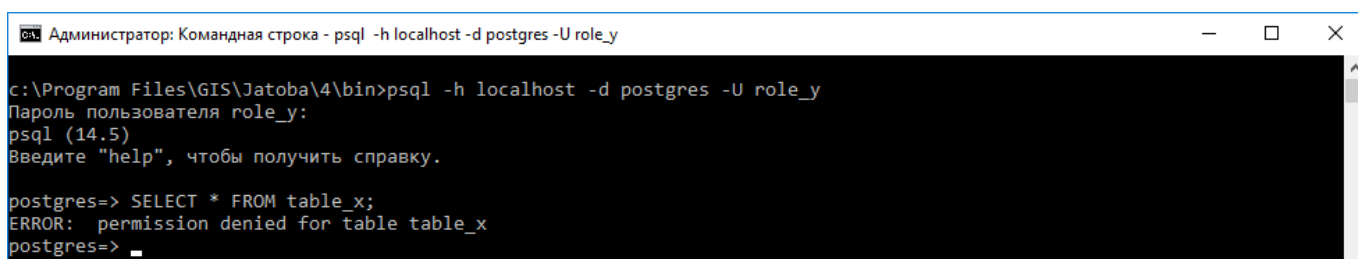
```
Администратор: Командная строка - psql -h localhost -d postgres -U postgres
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U postgres
Пароль пользователя postgres:
psql (14.5)
Введите "help", чтобы получить справку.

postgres=# SELECT * FROM table_x;
ERROR:  permission denied for table table_x
postgres=#
```

Рисунок 3.8 – Окно ошибки получения доступа Superuser к защищаемой таблице «table_x»

6) Пользователь «role_y» при попытке доступа к защищаемой таблице «table_x» получает отказ на выполнение операции:

```
SELECT * FROM table_x;
```



```
Администратор: Командная строка - psql -h localhost -d postgres -U role_y
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U role_y
Пароль пользователя role_y:
psql (14.5)
Введите "help", чтобы получить справку.

postgres=> SELECT * FROM table_x;
ERROR:  permission denied for table table_x
postgres=> _
```

Рисунок 3.9 – Окно попытки доступа к защищаемой таблице «table_x» пользователем «role_y»

7) Пользователь «dv_owner» устанавливает разрешение на работу с таблицей «table_x» пользователю «role_y»:

```
SELECT jdv.jdv_set_perm('table_x', 'role_y');
```

```

Администратор: Командная строка - psql -h localhost -d postgres -U dv_owner
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U dv_owner
Пароль пользователя dv_owner:
psql (14.5)
Введите "help", чтобы получить справку.

postgres=> SELECT jdv.jdv_set_perm('table_x', 'role_y');
 jdv_set_perm
-----
 t
(1 строка)

postgres=>
  
```

Рисунок 3.10 – Окно предоставления доступа пользователю «role_y» к защищаемой таблице «table_x»

8) Пользователь «role_y» успешно выполняет операцию:

```
SELECT * FROM table_x;
```

```

Администратор: Командная строка - psql -h localhost -d postgres -U role_y
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U role_y
Пароль пользователя role_y:
psql (14.5)
Введите "help", чтобы получить справку.

postgres=> SELECT * FROM table_x;
 i | j
----+---
(0 строк)

postgres=> _
  
```

Рисунок 3.11 – Окно получения доступа к защищаемой таблице

9) Пользователь «roleAIB» успешно просматривает таблицу защищаемых объектов «jdv.jdv_table»:

```
SELECT * from jdv.jdv_table;
```

```

Администратор: Командная строка - psql -h localhost -d postgres -U roleaib
c:\Program Files\GIS\Jatoba\4\bin>psql -h localhost -d postgres -U roleaib
Пароль пользователя roleaib:
psql (14.5)
Введите "help", чтобы получить справку.

postgres=> SELECT * from jdv.jdv_table;
 table_name | table_oid | role_name | role_oid
-----+-----+-----+-----
 table_x   |    16966 | owner    |    16963
 table_x   |    16966 | role_y   |    16964
(2 строки)

postgres=>
  
```

Рисунок 3.12 – Окно получения доступа к списку защищаемых таблиц

4. ОПИСАНИЕ ОПЕРАЦИЙ

4.1. Управление списком защищаемых таблиц

Роли dv_owner доступны функции управления списком защищаемых таблиц:

- 1) добавление в список защищаемых объектов «jdv.jdv_table» защищаемой таблицы:

```
jdv.jdv_add_table(table_name)
```



Объекты, находящиеся в этом списке, нельзя переименовать и удалить через команду DROP и DROP OWNED.

- 2) удаление из списка защищаемых объектов таблицы:

```
jdv.jdv_remove_table(table_name)
```

- 3) установка разрешения пользователю (роли) на работу с защищаемой таблицей:

```
jdv.jdv_set_perm(table_name, role_name)
```

- 4) отмена разрешения на работу с защищаемой таблицей пользователю (роли):

```
jdv.jdv_reset_perm(table_name, role_name)
```

4.2. Управление списком защищенных ролей

Роли dv_owner также доступны функции управления списком защищенных ролей:

- 1) добавление роли в список защищаемых:

```
jdv.jdv_add_role(role_name)
```



При добавлении таблиц и установке разрешений (через функции jdv_add_table и jdv_set_perm) роль-владелец или роль-пользователь добавляются в список защищаемых автоматически, поэтому эту функцию использовать в данном случае не требуется.

При выполнении данной функции заданная роль включается в группу dv_group.

- 2) удаление роли из списка защищаемых:

```
jdv.jdv_remove_role(role_name)
```



При удалении таблиц и отмене разрешений (через функции `jdv_remove_table` и `jdv_reset_perm`) роли из списка защищаемых автоматически не удаляются. Это связано с тем, что данная роль может иметь защищаемые объекты в других базах данных.

При выполнении данной функции заданная роль исключается из группы `dv_group`.

4.3. Управление функциями мониторинга

Ролям «dv_owner» и «dv_secanalyst» доступны функции мониторинга:

- 1) просмотр списка защищаемых объектов:

```
SELECT * from jdv.jdv_table where role_name = 'o w n e r'
```

- 2) просмотр списка разрешений:

```
SELECT * from jdv.jdv_table where role_name <> 'o w n e r'
```

4.3.1. Определение событий безопасности подлежащих регистрации компонентом JDV

При выполнении пользователями операций (успешных и неуспешных) в файл журнала `postgresql` записываются информационные сообщения. Компонентом JDV обеспечивается логирование следующих событий:

- 1) создание, изменение, удаление правила.

Например, сообщение об успешном добавлении таблицы `table1` в список защищаемых таблиц:

```
jdv * set * jdv_add_table('table1')
```

- 2) успешная попытка доступа пользователя к таблице/таблицам.

Например, сообщение об успешном получении ролью `role1` доступа к таблицам `schema1.table1` и `schema.table2` (роли `role1` ранее назначены разрешения на доступ к этим таблицам):

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
jdv * success * role1: schema1.table1
```

- 3) неуспешная попытка доступа пользователя к таблице/таблицам.

Например, сообщение о неуспешной попытке доступа роли role1 к таблицам schema1.table1 и schema.table2 (роли role1 ранее не назначены разрешения на доступ к этим таблицам):

```
jdv * prohibit * role1: schema1.table1
```

4.3.2. Управление регистрацией событий безопасности (логированием) компонента JDV

По умолчанию логирование включено для всех защищаемых таблиц и всех ролей, для схем не задано. Для выполнения логирования по событиям 2) и 3) требуется указать необходимые схемы, в которых находятся защищаемые объекты.

Компонент JDV позволяет настраивать логирование от имени и с правами роли «dv_owner». При этом доступны функции управления логированием ролей, объектов и схем:

- 1) функции для управления логированием ролей:
- добавление в логирование действий над объектами JDV ролей role1, role2.

```
jdv.jdv_log_include_role('role1','role2' | '*')
```

При этом возможно добавление как одной, так и нескольких ролей, а также добавление всех возможных ролей, установив параметр '*'.

- исключение из логирования действий над объектами JDV ролей role1, role2.

```
jdv.jdv_log_exclude_role('role1','role2' | '*')
```

При этом возможно исключение как одной, так и нескольких ролей, а также исключение всех возможных ролей, установив параметр '*'.

- 2) функции для управления логированием объектов.
- добавление в логи действий над объектами schema1.table1, schema1.table2

```
jdv.jdv_log_include_object('schema1.table1','schema1.table2' | '*')
```

Возможно добавление в логирование как одного, так и нескольких объектов, а также добавление всех возможных объектов установив параметр '*'.



По умолчанию для всех защищаемых таблиц логирование включено.

– исключение из логирования действий над объектами `schema1.table1`, `schema1.table2`

```
jdv.jdv_log_exclude_object('schema1.table1','schema1.table2' | '*')
```

Возможно исключение из логирования как одного, так и нескольких объектов, а также исключение всех возможных объектов, установив параметр '*'.

3) функции для управления логированием схем.

– добавление в логирование действий над всеми объектами схем `schema1`, `schema2`:

```
jdv_log_include_schema('schema1','schema2' | '*')
```

Возможно добавление в логирование действий как с одной, так и несколькими схемами, а также добавление всех возможных схем, установив параметр '*'.

– исключение из логирования действий над всеми объектами схем `schema1`, `schema2`:

```
jdv_log_exclude_schema('schema1','schema2' | '*')
```

Возможно исключение из логирования действий как над одной, так и над несколькими схемами, а также исключение всех возможных схем, установив параметр '*'.



По умолчанию для всех схем логирование отключено. Требуется задать необходимые схемы.

4.3.3. Управление местом хранения событий безопасности и политикой хранения

По умолчанию поток событий безопасности (логов) направляется в хранилище СУБД «Jatoba».

При необходимости, логирование событий доступа к объектам может также производиться в служебную таблицу `jdv_log`. Для этого имеются следующие функции:

- 1) функция для переключения места логирования:

```
jdv.jdv_log_dest('pglog' или 'pglog_table')
```

где:

- 'pglog' – поток логов компонента `jdv` направляется в хранилище СУБД «Jatoba»;
- 'pglog_table' – поток событий безопасности направляется в таблицу `jdv.jdv_log` и в хранилище СУБД «Jatoba».



Все служебные таблицы, в том числе таблица «`jdv.jdv_log`», доступны только для членов «`dv_owner`» и «`dv_secanalyst`».

В таблицу `jdv.jdv_log` записывается следующая информация:

- время начала транзакции;
- имя роли;
- имя объекта;
- статус доступа ('SUCCESS', 'PROHIBIT');
- текст команды (не более 300 символов).

- 2) функция для задания политики очистки таблицы «`jdv.jdv_log`»:

```
jdv_log_flush('параметр', значение)
```



Вызывать функцию может только `dv_owner`.

Принимает один из вариантов параметра:

- 'day' – дневная периодичность, в сутках;
- 'kb' – размерная периодичность, в килобайтах;
- 'rows' – размерная периодичность, в строках;
- 'now' – моментальная очистка;
- 'check' – просмотр установленной политики очистки;
- 'disable' – отмена политик очистки.

Параметры количества периодичности – 'day', 'size_kb', 'rows' – целое число.

Выставление новой политики очистки поверх старой отменяет старую политику автоматически.

Примеры применения команды `jdv_log_flush` приведены в таблице 1.

Таблица 1 – Примеры применения команды `jdv_log_flush`

Команда	Описание
<code>SELECT jdv_log_flush('day', 7)</code>	Все записи старше 7 дней удаляются. Проверка раз в 1 час
<code>SELECT jdv_log_flush('kb', 1024)</code>	При превышении таблицы размера в 1024 Кб самые старые логи удаляются. Проверка раз в 1 час
<code>SELECT jdv_log_flush('rows', 10000)</code>	При превышении таблицы размера в 10000 строк самые старые логи удаляются. Проверка раз в 1 час
<code>SELECT jdv_log_flush('disable')</code>	Отменяет политику очистки
<code>SELECT jdv_log_flush('check')</code>	Показывает текущую политику очистки: rows, 10000 / kb, 1024 / day, 7
<code>SELECT jdv_log_flush('now')</code>	Таблица полностью очищается

5. ВРЕМЕННОЕ ОТКЛЮЧЕНИЕ КОМПОНЕНТА

Компонент JDV может быть временно отключен, например, при выполнении обновления СУБД «Jatoba» до версии 18 включительно.

Для временного отключения компонента JDV необходимо проделать следующие шаги:

1) подключиться к СУБД ролью «dv_owner» и выполнить команду деактивации компонента:

```
SELECT jdv.jdv_deactivate();
```

2) подключиться к СУБД суперпользователем для выполнения любых операций без ограничений, например, выполнения обновления компонента (см. документ «Руководство по обновлению» 643.72410666.00067-07 93 01);

3) подключиться ролью dv_owner и выполнить команду активации компонента:

```
SELECT jdv.jdv_activate();
```



При отключенном компоненте нельзя использовать функции расширения.

6. УДАЛЕНИЕ КОМПОНЕНТА

Для полного удаления компонента JDV необходимо выполнить следующие действия:

- 1) подключиться ролью dv_owner и выполнить команду деактивации компонента:

```
SELECT jdv.jdv_deactivate();
```

- 2) подключиться к серверу под суперпользователем и выполнить:

```
ALTER ROLE ALL RESET session_preload_libraries;
```

- 3) удалить расширение командой:

```
DROP EXTENSION jdv;
```

- 4) удалить роли dv_group, dv_owner, dv_acctmgr и dv_secanalyst:

```
DROP ROLE dv_group, dv_owner, dv_acctmgr, dv_secanalyst;
```

- 5) удалить или закомментировать в конфигурационном файле postgresql.conf загрузку компонента JDV:

```
#shared_preload_libraries = jdv
```



Для удаления компонента JDV недостаточно выполнения команды –
DROP EXTENSION "jdv"

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

SQL	–	Structured Query Language — язык структурированных запросов
БД	–	База данных
ОС	–	Операционная система
СУБД	–	Система управления базами данных
ЭВМ	–	Электронно-вычислительная машина

Лист регистрации изменений

Дата внесения изм: