

УТВЕРЖДЕН
643.72410666.00067-07 95 01-ЛУ

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ «ЈАТОВА»

Руководство администратора

643.72410666.00067-07 95 01

Листов 237

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

АННОТАЦИЯ

Документ представляет собой руководство администратора системы управления базами данных «Jatoba» (далее по тексту – СУБД, СУБД «Jatoba»).

Руководство администратора содержит следующие разделы:

- раздел 1, в котором приведены назначение и функции СУБД «Jatoba» и требования к среде функционирования СУБД;
- раздел 2, в котором приведен состав СУБД «Jatoba»;
- раздел 3, в котором подготовка к установке СУБД «Jatoba»;
- раздел 4, в котором описана настройка параметров СУБД «Jatoba»;
- раздел 5, в котором описаны основные операции в СУБД «Jatoba»;
- раздел 6, в котором приведены настройки безопасности СУБД «Jatoba»;
- раздел 7, в котором приведена инструкция по резервному копированию и восстановлению баз данных;
- раздел 8, в котором приведена инструкция по созданию отказоустойчивого кластера СУБД «Jatoba»;
- раздел 9, в котором приведено описание восстановления поврежденных WAL записей;
- раздел 10, в котором приведено описание алгоритма KNN;
- раздел 11, в котором описано применение сертифицированных ОС для очистки памяти СУБД;
- раздел 12, в котором приведено описание компонента tsvector2;
- раздел 13, в котором приведено описание компонента gum;
- раздел 14, в котором приведено описание компонента xid64;
- раздел 15, в котором приведено описание компонента hunspell;
- раздел 16, в котором приведено описание функциональности поддержки автономных транзакций;

- раздел 17, в котором приведены сообщения об ошибках;
- раздел 18, в котором приведена информация о действиях после сбоев и возникших ошибках при эксплуатации СУБД «Jatoba»;
- приложение 1, в котором приведены значения полей из файла pg_hba.conf;
- приложение 2, которое содержит перечень событий СУБД с распределением по категориям безопасности.



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 Windows – «C:\Program Files\GIS\Jatoba\5\bin»;
- ОС Linux – «/usr/jatoba-5/bin».



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. Общие сведения о СУБД «Jatoba».....	8
1.1. Назначение СУБД «Jatoba».....	8
1.2. Функции СУБД «Jatoba».....	9
1.3. Требования к среде функционирования СУБД «Jatoba».....	9
1.3.1. Обеспечение контроля разрешений на действия пользователей, до и после идентификации и аутентификации	11
1.4. Ограничения при работе с СУБД «Jatoba».....	12
2. Состав СУБД «Jatoba»	13
2.1. Функциональные возможности и функциональные возможности по защите информации	16
2.1.1. Идентификация и аутентификация субъектов доступа и объектов доступа (ИАФ)	16
2.1.2. Управление доступом субъектов доступа к объектам доступа (УПД)	16
2.1.3. Регистрация событий безопасности (РСБ)	16
2.1.4. Обеспечение целостности информационной системы и информации (ОЦЛ).....	17
2.1.5. Обеспечение доступности информации (ОДТ)	17
2.1.6. Компоненты, расширяющие функции управления данными.....	18
3. Подготовка к установке СУБД «Jatoba»	20
4. Настройка параметров СУБД «Jatoba».....	21
4.1. Использование памяти	21
4.2. Использование дискового пространства	25
4.3. Использование ресурсов ядра.....	25
4.4. Настройка режима вакуумизации на основе стоимостных оценок	26
4.5. Настройка режима фоновой записи.....	28
4.6. Настройка режима асинхронного поведения	29
4.7. Параметры межсетевого взаимодействия	32
4.7.1. Основные параметры, связанные с настройкой сети	35
4.8. Настройка производительности СУБД «Jatoba»	37
4.9. Поддержка возможности включения принудительной очистки высвобождаемых блоков в файлах данных (Data wiping/Zeroing)	38
5. Основные операции в СУБД «Jatoba».....	39
5.1. Создание ролей в БД.....	39
5.2. Удаление ролей в БД.....	39
5.3. Создание БД	40
5.4. Удаление БД.....	41
5.5. Создание внешнего ТП	42
5.6. Удаление внешнего ТП.....	43
6. Настройка безопасности СУБД «Jatoba»	45
6.1. Идентификация и аутентификация субъектов доступа	45

6.1.1. Настройка конфигурационного файла pg_hba.conf	45
6.1.2. Настройка SSL	47
6.1.3. Настройка парольной политики. Компонент «SecurityProfile»	48
6.1.4. Удаление компонента «securityprofile»	86
6.1.5. Взаимодействие с компонентом управления кластером «ja_Dog»	87
6.1.6. Взаимодействие с компонентом контроля целостности «ja_CSum»	87
6.1.7. Взаимодействие с компонентом JDV	88
6.1.8. Взаимодействие с компонентом ja_hipe_cluster/Citus	89
6.2. Управление доступом субъектов доступа к объектам доступа	90
6.2.1. Блокирование и разблокирование учетных записей	90
6.2.2. Проверка установленных блокировок	103
6.2.3. Создание новых ролей, присвоение атрибутов и системных привилегий.....	105
6.2.4. Создание ролей при активированной парольной политике	110
6.2.5. Установка пароля пользователя	110
6.2.6. Блокирование сеанса доступа в СУБД после установленного времени бездействия (неактивности) пользователя.....	111
6.2.7. Прерывание текущих сессий в БД	112
6.3. Генератор паролей	113
6.3.1. Установка расширения pwgen.....	113
6.3.2. Генерация пароля	113
6.3.3. Генерация множества паролей	116
6.3.4. Удаление расширения «pwgen»	117
6.4. Регистрация событий безопасности СУБД «Jatoba»	117
6.4.1. Настройки регистраций событий безопасности СУБД «Jatoba» под управлением ОС Windows Server	117
6.4.2. Настройки регистрации событий безопасности СУБД «Jatoba» под управлением ОС семейства GNU/Linux.....	118
6.4.3. Компонент «pgAudit». Настройка расширенной регистрации событий безопасности	125
6.4.4. Компонент «pgauditlogtofile». Хранение событий безопасности в отдельном хранилище.....	135
6.4.5. Маскирование паролей.....	139
7. Резервное копирование и восстановление баз данных	142
7.1. Выгрузка кластера баз данных СУБД «Jatoba» в формате скрипта.....	142
7.2. Выгрузка определенной базы данных СУБД «Jatoba» в формате скрипта в файл	142
7.3. Восстановление базы данных СУБД «Jatoba» из файла архива	142
7.4. Создание резервной копии файлов СУБД «Jatoba»	143
8. Настройка отказоустойчивого кластера СУБД «Jatoba»	144
8.1. Настройка отказоустойчивого кластера СУБД «Jatoba» на ОС Windows Server	144
8.2. Настройка отказоустойчивого кластера СУБД «Jatoba» с использованием компонента «jaDog»	151
8.3. Поддержка асинхронной репликации данных между несколькими БД одного и того же типа	151

8.4. Поддержка задания временной задержки репликации между серверами	152
8.5. Развертывание отказоустойчивого кластера в приложении «Patroni» с СУБД «Jatoba»	153
9. Восстановление поврежденных WAL записей	155
10. Поиск ближайших соседей (KNN для B-Tree)	157
10.1. Форма SQL-запроса, для которых работает KNN	162
10.2. Влияние доработок по KNN на расширение btree_gist	162
10.3. Влияние доработок по KNN на системный каталог	163
10.3.1. Системная таблица pg_amop	164
10.3.2. Системная таблица pg_operator	165
10.3.3. Системная таблица pg_proc	166
11. Очистка памяти в СУБД	168
11.1. Очистка памяти СУБД средствами ОС	168
11.2. Очистка памяти СУБД встроенными средствами	168
12. Компонент tsvector2	171
12.1. Примеры использования компонента tsvector2	172
12.1.1. Функция «to_tsvector2». Преобразование текста в tsvector2	172
12.1.2. Индексация с использованием tsvector	173
12.1.3. Поиск с использованием tsvector2	174
12.1.4. Функции «jsonb_to_tsvector2» и «json_to_tsvector2». Преобразование json и jsonb в tsvector2	176
12.1.5. Функция «array_to_tsvector2». Преобразования из массива в строку	177
12.1.6. Функция «tsvector2_to_array». Преобразование из строки в массив	177
12.1.7. Функция «tsvector2_stat». Получение статистики по лексемам	177
12.1.8. Функции «tsvector2_update_trigger» и «tsvector2_update_trigger_column»	178
13. Компонент RUM	181
13.1. Общие операторы	181
13.2. Классы операторов	181
13.3. Примеры использования	183
13.3.1. Оператор rum_tsvector_ops	183
13.3.2. Оператор rum_anyarray_ops	184
14. Компонент xid64	185
15. Компонент hunspell	186
15.1. Установка расширения	187
15.2. Добавление словаря	188
15.3. Взаимодействие с компонентом tsvector2	188
15.4. Удаление расширения	189
15.5. Решение проблем	190
15.5.1. Словарь не найден	190
15.5.2. Ошибка прав доступа	190
16. Автономные транзакции	191

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

17. Сообщения об ошибках	193
18. Действия после сбоев и ошибок эксплуатации СУБД «Jatoba»	207
18.1. Временная блокировка пользователей СУБД и суперпользователя.....	207
18.2. Блокировка суперпользователя СУБД.....	209
18.3. Сбой инициализация расширения «securityprofile»	210
18.4. Ошибки создания пользователя.....	211
18.5. Ошибки, возникающие при использовании профиля парольных политик «securityprofile»	211
18.6. Ошибка авторизации.....	219
18.7. Компонент «ja_seceventlog». Ошибка загрузки библиотеки.....	220
18.8. Контактные данные службы технической поддержки.....	221
18.8.1. Версия изделия	221
Приложение 1	223
Приложение 2	225
Термины и определения	232
Перечень сокращений.....	234

1. ОБЩИЕ СВЕДЕНИЯ О СУБД «JAТОВА»

1.1. Назначение СУБД «Jatoba»

СУБД «Jatoba» является программным средством, предназначенным для создания и управления реляционными базами данных (БД) на базе электронно-вычислительных машин (ЭВМ) под управлением операционных систем (ОС), представленных в таблице 1.1.

Таблица 1.1 – Поддерживаемые ОС

№	Наименование ОС	Серверная часть	Клиентская часть	Docker (ver.)	Сертификат ФСТЭК	
					№ серт.	Дата выдачи
1	Windows 10	X	X	—	—	—
2	Windows 11	X	X	—	—	—
3	Windows Server 2016	X	X	—	—	—
4	Windows Server 2019	X	X	—	—	—
5	Windows Server 2022	X	X	—	—	—
6	Astra Linux 1.7 Special Edition Смоленск (x86-64)	X	X	25.0.5	2557	30.01.2012
7	Astra Linux 1.8 (x86-64)	X	X	25.0.5	—	—
8	Astra Linux 2.12 Common Edition Орел (x86-64)	X	X	—	—	—
9	Debian 10	X	X	24.0.2	—	—
10	Debian 11	X	X	24.0.2	—	—
11	Debian 12	X	X	24.0.2	—	—
12	АЛТ 8 СП	X	X	27.1.1	3866	10.08.2018
13	АЛТ 10 СП	X	X	27.1.1	3866	10.08.2018
14	АЛТ 9.1 Server	X	X	—	—	—
15	АЛТ 10 Server	X	X	23.0.1	—	—
16	Ubuntu 20.04	X	X	24.0.2	—	—
17	Ubuntu 22.04	X	X	24.0.2	—	—
18	Ubuntu 24.04	X	X	24.0.2	—	—
19	ОСНОВА2	X	X	20.10.5	4381	31.03.2021
20	РЕД ОС 7.3 Муром	X	X	25.0.7	4060	12.01.2019
21	РЕД ОС 8	X	X	—	—	—
22	РОСА 7.9	X	X	—	—	—
23	РОСА 12.4	X	X	—	—	—
24	RedHat Enterprise Linux 8	X	X	—	—	—
25	Oracle Linux 8.4	X	X	—	—	—

1.2. Функции СУБД «Jatoba»

СУБД «Jatoba» реализует следующие функциональные возможности:

- управление данными во внешней памяти;
- управление данными в оперативной памяти;
- выполнение запросов (DDL/DML);
- управление транзакциями;
- журнализация изменений, резервное копирование и восстановление базы

данных после сбоев, репликация.

СУБД «Jatoba» в дополнение к стандартным возможностям управления базами данных, реализует следующие функции:

- хранение пространственных, географических и геометрических данных, поддержка запросов к ним и управление ими;
- синтаксическая совместимость с распространенными PL/SQL Oracle;
- расширенные возможности секционирования больших таблиц;
- протоколирование, анализ и контроль выполнения команд манипулирования данными (DDL/DML);
- сбор журналов аудита всех операций и загрузка конфигураций в СУБД;
- работа в составе отказоустойчивого кластера с механизмом переключения нагрузки на основной узел кластера;
- защита от несанкционированного изменения конфигурационных файлов;
- единый пользовательский интерфейс для управления конфигурациями компонентов и просмотра их состояния СУБД.

1.3. Требования к среде функционирования СУБД «Jatoba»

СУБД «Jatoba» устанавливается на ЭВМ с процессорами, имеющими архитектуру x86, x86-64 и AMD64, удовлетворяющие следующим аппаратным требованиям, указанным в таблице 1.2.

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

Таблица 1.2 – Программные и аппаратные требования к средствам вычислительной техники, на которых функционируют клиентская и серверная часть СУБД

Параметр	Характеристика	Серт. ОС
Требования к аппаратному обеспечению сервера СУБД		
ОЗУ	Не менее 2 Гб	
Свободный объем жесткого диска	Минимальный объем от 40 Гб Рекомендуемый объем от 100 Гб	
Устройства видео вывода	Монитор и видеоадаптер с поддержкой VGA и разрешением 800х600 или выше	
Тип процессора и минимальная тактовая частота процессора	64-разрядный процессор Intel или AMD 3 ГГц или больше	
Минимальное количество ядер	4	
Максимальное количество ядер	256	
Устройства ввода-вывода	Стандартные 105-клавишная клавиатура и манипулятор «мышь» с USB, либо PS/2-интерфейсами	
Адаптер Ethernet	100 Мбит/с	
Требования к аппаратному обеспечению АРМ управления		
ОЗУ	Не менее 4 Гб	
Свободный объем жесткого диска	От 3 Гб	
Устройства видео вывода	Монитор и видеоадаптер с поддержкой VGA и разрешением 800х600 или выше	
Тип процессора и минимальная тактовая частота процессора	64-разрядный процессор Intel или AMD Рекомендуемая частота: 2.4 ГГц или больше	
Устройства ввода-вывода	Стандартные 105-клавишная клавиатура и манипулятор «мышь» с USB-интерфейсами, либо PS/2 интерфейсами	
Адаптер Ethernet	100 Мбит/с	
Требования к программному обеспечению сервера		
Операционная система	Требования приведены в таблице 1.1	
Требования к программному обеспечению АРМ управления		
Операционная система	Требования приведены в таблице 1.1	
Требования к аппаратному обеспечению сервера Jatoba data safe		
ОЗУ	Не менее 2 Гб	
Свободный объем жесткого диска	Минимальный объем от 40 Гб Рекомендуемый объем от 100 Гб	
Устройства видео вывода	Монитор и видеоадаптер с поддержкой HD и Full HD	
Тип процессора и минимальная тактовая частота процессора	64-разрядный процессор Intel или AMD 3 ГГц или больше	
№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____

Параметр	Характеристика	Серт. ОС
Минимальное количество ядер	4	
Устройства ввода-вывода	Стандартные 105-клавишная клавиатура и манипулятор «мышь» с USB, либо PS/2 интерфейсами	
Адаптер Ethernet	100 Мбит/с	
Требования к программному обеспечению сервера Jatoba data safe		
Поддерживаемые платформы	• win-x86;	—
	• win-x64;	—
	• linux-x64	X
СУБД	Защищенная система управления базами данных «Jatoba»	
Веб-сервер	IIS 10	—
	Nginx	X
Компоненты	ASP.NET Core 6.0 Runtime (v6.0.1) – Windows Hosting Bundle Installer	—
Internet браузер	• Google Chrome;	X
	• Яндекс.Браузер;	X
	• Chromium;	X
	• Mozilla Firefox;	X
	• Opera;	X
	• Microsoft Edge	—

1.3.1. Обеспечение контроля разрешений на действия пользователей, до и после идентификации и аутентификации

На уровне информационной системы для пользователей СУБД должно быть запрещено вносить несанкционированные изменения в объекты СУБД без идентификации и аутентификации.

Со стороны СУБД «Jatoba» контроль разрешений на действия пользователей, до и после идентификации и аутентификации осуществляется следующими методами:

- 1) В соответствии с Условиями эксплуатации отраженными в п. 3.12.10 Формуляра 643.72410666.00067-07 30 01, запрещено применения метода аутентификации «Trust», который предполагает, что любой подключающийся к серверу пользователь авторизован для доступа к базе данных вне зависимости от указанного имени пользователя базы данных.

- 2) Экосистема компонентов СУБД «Jatoba» позволяет выполнять централизованное управление всеми установками СУБД компонентом пользовательского веб-интерфейса для администраторов «Jatoba data safe».
- 3) Компонент пользовательского веб-интерфейса для администраторов «Jatoba data safe» при использовании функциональных возможностей разделов:
- раздел «Роли БД» (DB roles) предназначен для управления учетными записями пользователей и в частности назначения групповых ролей, что определяет набор прав и привилегий пользователя в СУБД;
 - разделы «Анализ рисков» (UserRisk) и «Матрица доступа» (Access matrix) позволяют контролировать назначаемые атрибуты и системные привилегии пользователям СУБД;
 - подраздел «Список событий» (Event List) аккумулирует события СУБД со всех установок СУБД;
 - раздел «Уведомления» (Notifications) позволяет поставить под контроль любые действия пользователей и администраторов СУБД.

1.4. Ограничения при работе с СУБД «Jatoba»

При использовании СУБД «Jatoba» необходимо учитывать следующие ограничения:

- при эксплуатации кластера с использованием СУБД «Jatoba» необходимо воздержаться от создания вложенных табличных пространств БД. Использование вложенных табличных пространств может привести к возникновению ошибки при выполнении процедуры восстановления/синхронизации резервных узлов кластера.

2. СОСТАВ СУБД «ЯТОВА»

В состав СУБД «Jatoba» входят компоненты, указанные в таблице 2.1.

Таблица 2.1 – Компоненты входящие в состав СУБД «Jatoba»

№	Наименование	Описание	J4		J5		J6	
			Дист. ¹⁾	Обр.к. ²⁾	Дист. ¹⁾	Обр.к. ²⁾	Дист. ¹⁾	Обр.к. ²⁾
1	ядро СУБД		X	X	X	X	X	X
	pwgen	генератор паролей	—	—	X	X	X	X
	—	маскирование паролей	—	—	X	X	X	X
	KNN	поиск ближайших соседей	—	—	X	X	X	X
	xid64	компонент xid64	—	—	X	X	X	X
	ja_Compression	сжатие данных на уровне страниц	—	—	—	—	X	X
	WAL Recovery	восстановление поврежденных WAL записей	—	—	X	X	X	X
2	jaDog	компонент управления режимом работы узлов кластера	X	X	X	X	X	X
3	JDV (Jatoba data vault)	компонент контроля субъектов доступа	X	X	X	X	X	X
4	pgBadger	компонент формирования отчетов по журналам СУБД	X	X	X	X	X	X
5	pg_ProBackup	компонент расширенного резервного копирования	X	X	X	X	X	X
6	pg_Task	компонент планирования заданий СУБД	X	X	X	X	X	X
7	pg_Profile	компонент формирования отчетов производительности СУБД	X	X	X	X	X	X
8	JDS (Jatoba data safe)	компонент пользовательского веб-интерфейса для администраторов	X	—	X	—	X	—
9	ja_Sync_Ldap	компонент синхронизации учетных записей со службами каталогов	X	X	X	X	X	X
10	PlsPgSQL	компонент обфускации кода PL/pgSQL	X	—	X	—	X	—
11	ja_Hipe_Cluster	компонент высокопроизводительного кластера	X	X	X	X	X	X
12	ja_Log	компонент централизованного сбора записей событий СУБД	X	—	X	—	X	—
13	1c_support	компонент поддержки платформы 1C	X	X	X	X	X	X
14	fasttrun	компонент совместимости с 1C	X	X	X	X	X	X
15	fulleq	компонент совместимости с 1C	X	X	X	X	X	X
16	mchar	компонент совместимости с 1C	X	X	X	X	X	X
17	online_analyze	компонент совместимости с 1C	X	X	X	X	X	X
18	plantuner	компонент совместимости с 1C	X	X	X	X	X	X
19	ja_CSum	компонент контроля целостности	X	X	X	X	X	X
20	jaPooler	компонент балансировки подключений пользователей к СУБД	X	—	X	—	X	—

№	Наименование	Описание	J4		J5		J6	
			Дист. ¹⁾	Обр.к. ²⁾	Дист. ¹⁾	Обр.к. ²⁾	Дист. ¹⁾	Обр.к. ²⁾
21	Oracle_FDW (Foreign data wrapper for oracle)	компонент доступа к данным СУБД Oracle	X	X	X	X	X	X
22	OraFCE (Oracle function compatibility extension)	компонент совместимости с СУБД Oracle	X	X	X	X	X	X
23	pg_Variables	компонент совместимости с системой глобальных переменных СУБД Oracle	X	X	X	X	X	X
24	SQL_Firewall	компонент выявления и предотвращения исполнения нетипичных SQL-запросов	X	X	X	X	X	X
25	JCS (Jatoba crypto access storage)	компонент сокрытия информации в файлах данных СУБД	X	X	X	X	X	X
26	pgSQL-HTTP	компонент формирования HTTP/HTTPS запросов из СУБД	X	X	X	X	X	X
27	TDS_FDW	компонент поддержки платформы Microsoft SQL Server	—	—	X	X	X	X
28	pgAudit	компонент расширенного журналирования событий СУБД	X	X	X	X	X	X
29	pgauditlogtofile	хранение событий безопасности в отдельном хранилище	—	—	X	X	X	X
30	PostGIS	компонент работы с географическими данными	X	X	X	X	X	X
31	PTrack	компонент расширенного резервного копирования	X	X	X	X	X	X
32	SecurityProfile	компонент управления парольными политиками пользователей СУБД	X	X	X	X	X	X
33	ja_Plan_Manager	компонент создания планов запросов в базах данных (БД), их оптимизации и экспорта в БД	X	X	X	X	X	X
34	pg_store_plans	контроль выполненных планов запросов	—	—	X	X	X	X
35	pg-hint-plan	компонент корректировки запросов	—	—	X	X	X	X
36	ja_Container	СУБД «Jatoba» в контейнере	—	X	—	X	—	X
37	node_exporter	компонент сбора аппаратных и программных показателей работы GNU/Linux	—	—	X	—	X	—
38	postgres_exporter	компонент сбора метрик СУБД	—	—	X	—	X	—
39	sql_exporter	SQL экспортёр. Компонент сбора расширенных метрик СУБД	—	—	X	—	X	—
40	prometheus	компонент мониторинга различных программных систем и сервисов Prometheus	—	—	X	—	X	—

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

№	Наименование	Описание	J4		J5		J6	
			Дист. ¹⁾	Обр.к. ²⁾	Дист. ¹⁾	Обр.к. ²⁾	Дист. ¹⁾	Обр.к. ²⁾
41	Alertmanager	компонент управления и обработки оповещений в системе мониторинга Prometheus	—	—	X	—	X	—
42	—	Работа СУБД «Jatoba» в режиме ЗПС в ОС Astra Linux	X	—	X	—	X	—
43	gis-cryptoplatform	библиотека «ГИС»	X	—	X	—	X	—
44	pg_ulid	компонент поддержки лексографического идентификатора	—	—	X	X	X	X
45	ja_Seceventlog	компонент записи событий информационной безопасности	—	—	X	X	X	X
46	rum	компонент поддерживающий обратный индекс с хранением позиционной информации и полнотекстовый поиск	—	—	X	X	X	X
47	pg_repack	компонент реорганизации таблицы с минимальными блокировками	—	—	X	X	X	X
48	osnova-digsig-key	работа СУБД Jatoba в режиме ЗПС в ОС ОСНОВА	X	—	X	—	X	—
49	tsvector2	компонент полнотекстового поиска в БД	—	—	X	X	X	X
50	ja_Similar	компонент для полнотекстового поиска и определения похожих текстов	—	—	X	X	X	X
51	ja_Inventory	компонент инвентаризации СУБД	X	—	X	—	X	—
52	ja_tune	Генератор конфигурационного файла	—	—	—	—	X	—
53	hunspell	Свободная библиотека для проверки орфографии и морфологического анализа. Компонент «hunspell»	—	—	X	X	X	X
54	ja_anonymizer	Маскирование данных. Компонент «ja_anonymizer»	—	—	—	—	X	X
55	wal-g	Архивация и восстановление данных. Компонент «wal-g»	—	—	—	—	X	X

Примечание:

- Дистрибутив.
- Образ контейнера.

2.1. Функциональные возможности и функциональные возможности по защите информации

2.1.1. Идентификация и аутентификация субъектов доступа и объектов доступа (ИАФ)

Функциональные возможности по идентификации и аутентификации пользователей (ИАФ.1) выполняются средствами СУБД и описаны в текущем документе:

Руководство администратора 643.72410666.00067-07 95 01

Установление характеристик пароля (ИАФ.4) обеспечивает компонент SecurityProfile, функциональные возможности которого описаны в текущем документе:

Руководство администратора 643.72410666.00067-07 95 01

2.1.2. Управление доступом субъектов доступа к объектам доступа (УПД)

Меры по управлению доступом субъектов доступа выполняются штатными средствами СУБД, описанными в данном документе:

Руководство администратора 643.72410666.00067-07 95 01

Функциональные возможности СУБД по заведению учетных записей пользователей (УПД.1) обеспечивает компонент «ja_Sync_LDAP». Описание компонента представлено в документе:

Руководство по настройке. Часть 8. Синхронизация учетных записей служб каталогов и СУБД. Компонент «ja_Sync_LDAP» 643.72410666.00067-07 98 01-08

Функциональной возможностью по назначению минимально необходимых прав и привилегий пользователям и администраторам (УПД.5) обладает компонент «Jatoba data vault». Описание компонента представлено в документе:

Руководство по настройке. Часть 2. Контроль субъектов доступа. Компонент «Jatoba data vault» 643.72410666.00067-07 98 01-02

2.1.3. Регистрация событий безопасности (РСБ)

Регистрация событий безопасности выполняется средствами СУБД. Для расширенного журналирования используется компонент pgAudit, функциональные возможности которого описаны в текущем документе:

Руководство администратора 643.72410666.00067-07 95 01

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

Управление передачей событий безопасности выполняется компонентом «ja_Log», функциональные возможности которого описаны в документе:

Руководство по настройке. Часть 12. Централизованный сбор записей событий СУБД. Компонент «ja_Log» 643.72410666.00067-07 98 01-12

Компонент «Jatoba data safe» осуществляет управление передачи событий, а также обеспечивает:

- централизованное автоматизированное управление сбором, записью и хранением информации о событиях безопасности (РСБ.3);
- возможность просмотра и анализа информации о действиях отдельных пользователей в информационной системе (РСБ.8);
- постоянный и периодический контроль за состоянием целевых СУБД, уровнем их безопасности;
- управление кластером.

Подробное описание компонента приведено в документе:

Руководство по настройке. Часть 7. Пользовательский веб-интерфейс для администраторов. Компонент «Jatoba data safe» 643.72410666.00067-07 98 01-07

2.1.4. Обеспечение целостности информационной системы и информации (ОЦЛ)

Контроль целостности собственных компонентов по контрольным суммам осуществляется динамически в процессе работы СУБД (ОЦЛ.1) и обеспечивается компонентом «ja_CSum». Описание компонента приведено в документе:

Руководство по настройке. Часть 14. Компонент контроля целостности «ja_CSum» 643.72410666.00067-07 98 01-14

2.1.5. Обеспечение доступности информации (ОДТ)

Обеспечение доступности информации достигается периодическим резервным копированием информации (ОДТ.4) и обеспечением возможности восстановления информации (ОДТ.5). Функции по защите информации выполняет компонент «pg_ProBackup». Описание приведено в документе:

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

Руководство по настройке. Часть 4. Расширенное резервное копирование. Компонент «pg_ProBackup» 643.72410666.00067-07 98 01-04

Обеспечение доступности информации также достигается кластеризацией серверов БД (ОДТ.6). Кластеризация БД может быть выполнена компонентом «jaDog». Описание компонента приведено в документе:

Руководство по настройке. Часть 1. Управление режимом работы узлов кластера. Компонент «jaDog» 643.72410666.00067-07 98 02-01

2.1.6. Компоненты, расширяющие функции управления данными

Прочие компоненты, расширяющие функциональные возможности СУБД, описаны в документах:

- *«Руководство по настройке. Часть 3. Формирование отчетов по журналам СУБД. Компонент «pgBadger» 643.72410666.00067-07 98 01-03»;*
- *«Руководство по настройке. Часть 5. Планирование заданий СУБД. Компонент «pg_Task» 643.72410666.00067-07 98 01-05»;*
- *«Руководство по настройке. Часть 6. Формирование отчетов производительности СУБД. Компонент «pg_Profile» 643.72410666.00067-07 98 01-06»;*
- *«Руководство по настройке. Часть 9. Обфускация кода PL/pgSQL. Компонент «PLsPgSQL» 643.72410666.00067-07 98 01-09»;*
- *«Руководство по настройке. Часть 10. Компрессия данных СУБД. Компонент «pg_Cryogen» 643.72410666.00067-07 98 01-10»;*
- *«Руководство по настройке. Часть 11. Высокопроизводительный кластер. Компонент «ja_Hipe_Cluster» 643.72410666.00067-07 98 01-11»;*
- *«Руководство по настройке. Часть 13. Поддержка платформы IC 643.72410666.00067-07 98 01-13»;*
- *«Руководство по настройке. Часть 15. Балансировка подключений пользователей к СУБД. Компонент «jaPooler» 643.72410666.00067-07 98 01-15»;*
- *«Руководство по настройке. Часть 16. Обеспечение работы с СУБД Oracle. 643.72410666.00067-07 98 01-16»;*

- *«Руководство по настройке. Часть 17. Выявление и предотвращение исполнения нетипичных SQL-запросов. Компонент «SQL_Firewall» 643.72410666.00067-07 98 01-17»;*
- *«Руководство по настройке. Часть 18. Соккрытие информации в файлах данных СУБД. Компонент «Jatoba crypto access storage» 643.72410666.00067-07 98 01-18»;*
- *«Руководство по настройке. Часть 19. Формирование HTTP/HTTPS запросов из СУБД. Компонент «pgSQL-HTTP» 643.72410666.00067-07 98 01-19»;*
- *«Руководство по настройке. Часть 20. Компонент «TDS_FDW» 643.72410666.00067-07 98 01-20»;*
- *«Руководство по настройке. Часть 21. Управление планами запросов. Компонент «ja_Plan_Manager» 643.72410666.00067-07 98 01-21»;*
- *«Руководство по настройке. Часть 30. Запись событий информационной безопасности. Компонент ja_seceventlog» 643.72410666.00067-07 98 01-30».*

3. ПОДГОТОВКА К УСТАНОВКЕ СУБД «JATOBA»

Комплект установочных файлов СУБД «Jatoba» с документацией поставляется заказчику на установочном компакт-диске.

Перед началом установки необходимо:

- проверить комплектность поставки СУБД «Jatoba» в соответствии с требованиями раздела 4 документа «Защищенная система управления базами данных «Jatoba». Формуляр» 643.72410666.00067-07 30 01;
- провести визуальный осмотр компакт-диска с дистрибутивными файлами и эксплуатационной документацией на предмет повреждений;
- выполнить проверку информации, записанной на компакт-диске, на возможность чтения и соответствия имен файлов и их контрольных сумм, указанным в перечне файлов, приведенном в Приложении 1 к документу «Защищенная система управления базами данных «Jatoba». Формуляр» 643.72410666.00067-07 30 01.

Контрольные суммы установочных файлов на дистрибутивном компакт-диске СУБД «Jatoba» получены с помощью программы фиксации и контроля исходного состояния программного комплекса «ФИКС» версии 2.0.2 (производитель ЗАО «ЦБИ-сервис») по алгоритму «Уровень-3».

Установка СУБД проводится в соответствии с документом «Защищенная система управления базами данных «Jatoba». Руководство по установке» 643.72410666.00067-07 97 01.

4. НАСТРОЙКА ПАРАМЕТРОВ СУБД «ЯТОВА»

В данном разделе описываются основные параметры конфигурации, которые влияют на работу БД.

4.1. Использование памяти

```
shared_buffers (integer)
```

Задаёт объем разделяемой памяти, которую сервер баз данных будет использовать для размещения буферов со страницами файлов данных. Данная память будет совместно использоваться всеми процессами СУБД.

По умолчанию значение `shared_buffers` = 128 Мб, но может быть меньше, если такой объем не поддерживается операционной системой (определяется в процессе инициализации директории данных служебной утилитой `initdb`). Значение этого параметра не может быть меньше 128 Кб (килобайт) (минимум зависит от величины `BLCKSZ` – размер блока данных в файле данных, по умолчанию 8 Кб). Для хорошей производительности требуются большие значения.

После настройки данного параметра для вступления его в действие требуется перезапуск сервера.

При использовании сервера с объемом ОЗУ 1 Гб наиболее оптимальным начальным значением `shared_buffers` будет 25% от объема памяти. Увеличение `shared_buffers` обычно требует увеличения `max_wal_size`, чтобы растянуть процесс записи большого объема новых или измененных данных на больший промежуток времени.

При использовании сервера с объемом ОЗУ меньше 1 Гб стоит ограничиться меньшим процентом ОЗУ, чтобы оставить достаточно места операционной системе.

```
huge_pages (enum)
```

Определяет, будут ли запрашиваться страницы большого размера (`huge pages`) из основной области разделяемой памяти.

При `huge_pages` = `try` (по умолчанию) сервер будет запрашивать выделение памяти страницами большого размера. Если сервер получит ошибку выделения памяти, то он вернется к стандартному поведению (выделение памяти страницами стандартного размера).

При `huge_pages = on` сервер не будет запущен, если в ОС не будет возможности выделять память страницами большого размера.

При `huge_pages = off` выделение памяти будет производиться страницами стандартного размера.



Данный параметр поддерживается только в ОС Linux и Windows. В остальных ОС значение `true` игнорируется.

В результате использования страниц большого размера уменьшаются риски промахов в TLB кэше и процессор тратит меньше времени на преобразование адресов, что приводит к увеличению быстродействия.

Для того, чтобы пользователь мог использовать страницы большого размера в ОС Windows, необходимо дать пользователю Windows, от имени которого работает СУБД «Jatoba», право блокировки страниц в памяти (Lock Pages in Memory в управлении групповой политикой в Windows).

```
temp_buffers (integer)
```

Задаёт максимальное число временных буферов для каждой сессии пользователя.

По умолчанию `temp_buffers = 8` (8Мб = 1024 буфера).

Данный параметр можно изменить до первого обращения к временным таблицам в рамках сессии пользователя. После изменить значение этого параметра для текущей сессии будет невозможно.

В рамках сессии временные буферы выделяются по мере необходимости до достижения предела, который задан параметром `temp_buffers`. Если сессия не задействует временные буферы, то для него хранятся только дескрипторы буферов, которые занимают около 64 байт (в количестве `temp_buffers`). Если буфер используется, он будет дополнительно занимать 8192 байта (или `BLCKSZ` байт, в общем случае).

```
max_prepared_transactions (integer)
```

Задаёт максимальное число транзакций, которые могут одновременно находиться в «подготовленном» состоянии.

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

По умолчанию `max_prepared_transactions = 0` отключается механизм подготовленных транзакций. Задать данный параметр можно только при запуске сервера.

Если нет необходимости использовать подготовленные транзакции, следует обнулить параметр, чтобы не допустить непреднамеренного создания подготовленных транзакций. Если подготовленные транзакции используются, то `max_prepared_transactions` должен быть не меньше, чем `max_connections` для подготовки транзакции в каждом сеансе.



Для ведомого сервера значение этого параметра должно быть больше или равно значению на ведущем. В противном случае, на ведомом сервере запросы будут запрещены.

```
work_mem (integer)
```

Задаёт объем памяти, который будет использоваться для внутренних операций сортировки и хэш-таблиц прежде чем будут задействованы временные файлы на диске.

По умолчанию `work_mem = 4` (4 Мб). В сложных запросах может одновременно выполняться несколько операций сортировки или хэширования, при этом, указанный объем памяти может использоваться в каждой операции перед тем, как данные начнут перемещаться во временные файлы.

Общий объем памяти может превосходить значение `work_mem`. Операции сортировки используются для `ORDER BY`, `DISTINCT` и соединений слиянием.

```
maintenance_work_mem (integer)
```

Задаёт максимальный объем памяти для операций обслуживания БД, таких как `VACUUM`, `CREATE INDEX` и `ALTER TABLE ADD FOREIGN KEY`.

По умолчанию `maintenance_work_mem (integer) = 64` (64 Мб). Увеличение данного значения может привести к ускорению операций очистки и восстановления БД из копии.

При работе автовакуума объем памяти выделяется `autovacuum_max_workers` один раз, не рекомендуется устанавливать значение по умолчанию слишком большим. Управлять объемом памяти для автовакуума предпочтительно отдельно, изменяя `autovacuum_work_mem`.



Для сбора идентификаторов мертвых кортежей VACUUM может использовать не более 1Гб памяти.

```
autovacuum_work_mem (integer)
```

Задаёт максимальный объём памяти, который будет использовать каждый рабочий процесс автовакуума.

При значении по умолчанию `autovacuum_work_mem = -1` объём определяется значением `maintenance_work_mem`. Данный параметр не влияет на поведение команды VACUUM, которая может выполняться в других контекстах. Задать этот параметр можно только в `postgresql.conf` или в командной строке при запуске сервера. Увеличение `autovacuum_work_mem` до большего значения не повлияет на количество обнаруженных удалённых записей, которые автовакуум собирает при сканировании таблицы.



Для сбора идентификаторов мертвых кортежей VACUUM может использовать не более 1Гб памяти.

```
max_stack_depth (integer)
```

Задаёт максимальную безопасную глубину стека для исполнителя.

По умолчанию `max_stack_depth = 2` (2 Мб). Значение выбрано с запасом, переполнение стека невозможно за исключением выполнения сложных функций. Изменить этот параметр могут только суперпользователи.

Данное значение рекомендуется ставить равным предельному размеру стека, ограниченному ядром (который устанавливается командой `ulimit -s` или аналогичной), за вычетом запаса в 1 Мб. Запас необходим для потенциально рекурсивных процедур.



При превышении значения `max_stack_depth` фактического предела ядра, функция с неограниченной рекурсией сможет вызвать экстренное завершение работы отдельного процесса сервера.

```
dynamic_shared_memory_type (enum)
```


Выбирает механизм динамической разделяемой памяти, который будет использоваться сервером:

- при `dynamic_shared_memory_type = posix` происходит выделение разделяемой памяти POSIX функцией `shm_open` (данный механизм ставится по умолчанию);
- при `dynamic_shared_memory_type = sysv` происходит выделение разделяемой памяти System V функцией `shmget`;
- при `dynamic_shared_memory_type = windows` происходит выделение разделяемой памяти в Windows;
- при `dynamic_shared_memory_type = mmap` происходит эмуляция разделяемой памяти через отображение в память файлов, хранящихся в каталоге данных;
- при `dynamic_shared_memory_type = none` происходит отключение этой функциональности.

4.2. Использование дискового пространства

```
temp_file_limit (integer)
```

Задаёт максимальный объём дискового пространства, который сможет использовать один процесс для временных файлов, например, при сортировке и хэшировании или для сохранения удерживаемого курсора. Транзакция, которая попытается превысить этот предел, будет отменена.

По умолчанию `temp_file_limit = -1` (-1Кб) означает, что предел отсутствует. Изменить данный параметр могут только суперпользователи.

Этот параметр ограничивает общий объём, который могут занимать в момент времени все временные файлы, которые задействованы в данном процессе СУБД.



Это не касается файлов явно создаваемых временных таблиц.

Ограничивается объём временных файлов, которые создаются неявно при выполнении запросов.

4.3. Использование ресурсов ядра

```
max_files_per_process (integer)
```

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

Задаёт максимальное число файлов, которые могут быть одновременно открыты каждым серверным подпроцессом.

По умолчанию `max_files_per_process = 1000` файлов. Задать этот параметр можно только при запуске сервера.

Если ядро реализует безопасное ограничение по процессам, то значение данного параметра можно не менять. На некоторых платформах ядро позволяет отдельному процессу открыть больше файлов, чем могут открыть несколько процессов одновременно. При возникновении ошибки «Too many open files», необходимо уменьшить значение параметра.

4.4. Настройка режима вакуумизации на основе стоимостных оценок

Во время выполнения команд `VACUUM` и `ANALYZE` система ведёт внутренний счётчик, в котором суммируется оцениваемая стоимость различных выполняемых операций ввода/вывода. При превышении накопленной стоимости `vacuum_cost_limit`, процесс, выполняющий эту операцию, отключается на время `vacuum_cost_delay`, после чего счётчик сбрасывается и процесс продолжается.

Данный подход реализован для снижения влияния этих команд на параллельную работу с базой, за счёт уменьшения нагрузки на подсистему ввода-вывода. Важно, чтобы команды меньше влияли на выполнение других операций с базой данных. Данным процессом могут управлять администраторы.

По умолчанию данный режим отключен для выполняемых вручную команд `VACUUM`. Для его включения нужно установить в `vacuum_cost_delay` ненулевое значение.

```
vacuum_cost_delay(integer)
```

Продолжительность времени, в течение которого будет простаивать процесс, превысивший предел стоимости.

По умолчанию `vacuum_cost_delay = 0` (0 мс) – задержка очистки отсутствует. При положительных значениях интенсивность очистки будет зависеть от стоимости.



Разрешение таймера `vacuum_cost_delay` должно быть кратно 10.

При настройке интенсивности очистки для `vacuum_cost_delay` выбираются небольшие значения (например, 10 или 20 мс). Для точного ограничения потребления ресурсов при очистке рекомендуется изменять другие параметры стоимости очистки.

```
vacuum_cost_page_hit (integer)
```

Примерная стоимость очистки буфера, оказавшегося в общем кэше. Содержит в себе блокировку пула буферов, поиск в хэш-таблице и сканирование содержимого страницы.

По умолчанию `vacuum_cost_page_hit = 1`.

```
vacuum_cost_page_miss (integer)
```

Примерная стоимость очистки буфера, который нужно прочитать с диска. Содержит в себе блокировку пула буферов, поиск в хэш-таблице, чтение требуемого блока с диска и сканирование его содержимого.

По умолчанию `vacuum_cost_page_miss = 10`.

```
vacuum_cost_page_dirty (integer)
```

Примерная стоимость очистки, при которой изменяется блок немодифицированный ранее. В данный параметр включается дополнительная стоимость ввода/вывода, связанная с записью измененного блока на диск.

По умолчанию `vacuum_cost_page_dirty = 20`.

```
vacuum_cost_limit (integer)
```

Общая стоимость, при накоплении которой процесс очистки будет выключаться.

По умолчанию `vacuum_cost_limit = 200`.



Некоторые операции могут устанавливать критические блокировки и должны завершаться как можно быстрее. Во время таких операций задержка очистки по стоимости не осуществляется, поэтому накопленная за это время стоимость может быть больше установленного предела.

Во избежание ненужных длительных задержек фактическая задержка вычисляется по формуле:

$$\frac{(\text{vacuum_cost_delay} * \text{accumulated_balance})}{\text{vacuum_cost_limit}} \leq \text{vacuum_cost_delay} * 4.$$

4.5. Настройка режима фоновой записи

В числе специальных процессов сервера есть процесс фоновой записи, задачей которого является осуществление записей новых или измененных («грязных») общих буферов на диск. При недостаточном количестве чистых общих буферов данный процесс записывает грязные буферы в файловую систему и помечает их как чистые. Процесс фоновой записи увеличивает общую нагрузку на подсистему ввода/вывода, так как может записывать изменяемую страницу несколько раз, хотя ее можно было бы записать один раз в момент контрольной точки.

```
bgwriter_delay (integer)
```

Задаёт задержку между раундами активности процесса фоновой записи. Во время раунда процесс осуществляет запись определенного количества загрязненных буферов. Затем данный процесс выключается на время `bgwriter_delay` (в миллисекундах) и так повторяется. Если в пуле не остается загрязненных буферов, он может быть неактивен более длительное время.

По умолчанию `bgwriter_delay = 200` (200мс). Задать параметр можно в `postgresql.conf` или в командной строке при запуске сервера.

```
bgwriter_lru_maxpages (integer)
```

Задаёт максимальное число буферов, которое сможет записать процесс фоновой записи за раунд активности.

По умолчанию `bgwriter_lru_maxpages = 100` (100 буферов).

При `bgwriter_lru_maxpages = 0` фоновая запись отключается. Задать параметр можно в `postgresql.conf` или в командной строке при запуске сервера.

```
bgwriter_lru_multiplier (floating point)
```

Число загрязненных буферов, записываемых в очередном раунде, которое зависит от количества новых буферов, требуемых серверным процессам в предыдущих раундах.

По умолчанию `bgwriter_lru_multiplier = 2.0`. Задать параметр можно в `postgresql.conf` или в командной строке при запуске сервера.

Значение `bgwriter_lru_multiplier` умножается на накопленное усредненное значение количества использованных буферов на предыдущих раундах и получается значение, равное количеству буферов для следующего раунда. Процесс фоновой записи пишет на диск и освобождает буферы до тех пор, пока число свободных буферов не достигнет целевого значения. Число буферов, которые записаны за раунд, ограничиваются параметром `bgwriter_lru_maxpages`.

```
bgwriter_flush_after (integer)
```

При большем количестве байт, которые записываются процессом фоновой записи, чем `bgwriter_flush_after`, сервер посылает команду ОС произвести запись этих данных в нижележащее хранилище. Это ограничивает объем «грязных» данных в страничном кэше ядра и уменьшает вероятность затормаживания при выполнении `fsync` в конце контрольной точки, или когда ОС сбрасывает данные на диск большими порциями в фоне.

По умолчанию в ОС Linux `bgwriter_flush_after = 512` (512 Кб). В других ОС `bgwriter_flush_after = 0`.

Параметр действует не на всех платформах и может принимать значение от 0 (управление отложенной записью отключается) до 2 Мб. Если `BLCKSZ` отличен от 8 Кб, максимальное значение корректируется пропорционально. Задать этот параметр можно только в `postgresql.conf` или в командной строке при запуске сервера.

4.6. Настройка режима асинхронного поведения

```
effective_io_concurrency (integer)
```

Задаёт допустимое число параллельных операций ввода/вывода, которые могут быть выполнены одновременно. Чем больше это число, тем больше операций ввода/вывода будет пытаться выполнить СУБД параллельно в отдельном сеансе. Допустимые значения

находятся в интервале от 1 до 1000, а нулевое значение отключает асинхронные запросы ввода/вывода.

Для магнитных носителей начальным значением этого параметра будет являться число отдельных дисков, составляющих массив RAID 0 или RAID 1, в котором размещена база данных. Если база данных часто обрабатывает множество запросов в различных сеансах, то при небольших значениях дисковый массив может быть полностью загружен. При увеличении этого значения при полной загрузке дисков, это приведет к увеличению нагрузки на процессор.

По умолчанию `effective_io_concurrency = 1`, где данный параметр поддерживается, и `effective_io_concurrency = 0` – в остальных. Значение можно переопределить для таблиц в определенном табличном пространстве, установив одноименный параметр табличного пространства.

```
max_worker_processes (integer)
```

Задаёт максимальное число фоновых процессов, которое можно запустить в текущей системе.

По умолчанию `max_worker_processes = 8`. Параметр можно задать только при запуске сервера.

Для ведомого сервера значение данного параметра должно быть больше или равно значению на ведущем. В противном случае на ведомом сервере не будут разрешены запросы.

```
max_parallel_workers_per_gather (integer)
```

Задаёт максимальное число рабочих процессов, которые могут запускаться одним узлом плана запроса Gather или Gather Merge (сбор результатов с рабочих процессов). Параллельные рабочие процессы берутся из пула процессов, который контролируется параметром `max_worker_processes`, в количестве, ограничиваемом значением `max_parallel_workers`. Запрошенное количество рабочих процессов может быть недоступно во время выполнения. В таком случае параметр будет выполняться с меньшим числом процессов, что может быть неэффективно.

По умолчанию `max_parallel_workers_per_gather = 2`.

При `max_parallel_workers_per_gather = 0` отключается параллельное выполнение запросов.

Параллельные запросы потребляют больше ресурсов, чем непараллельные, так как каждый рабочий процесс является отдельным процессом. Рекомендуется это учитывать, выбирая значение параметра, а также настраивая другие параметры, управляющие использованием ресурсов.

```
max_parallel_maintenance_workers (integer)
```

Задаёт максимальное число рабочих процессов, которые могут запускаться одной служебной командой. Параллельные рабочие процессы берутся из пула процессов, который контролируется параметром `max_worker_processes`, в количестве, ограничиваемом значением `max_parallel_workers`. Запрошенное количество рабочих процессов может быть недоступно во время выполнения. В таком случае служебная операция будет выполняться с меньшим числом процессов, чем ожидалось.

По умолчанию `max_parallel_maintenance_workers = 2`.

При `max_parallel_maintenance_workers = 0` отключается использование параллельных исполнителей служебными командами.

Параллельно выполняемые служебные команды не должны потреблять значительно больше памяти, чем равнозначные непараллельные операции. Это отличает их от параллельных запросов, при выполнении которых ограничения ресурсов действуют на отдельные рабочие процессы.

```
max_parallel_workers (integer)
```

Задаёт максимальное число рабочих процессов, которое система сможет поддерживать для параллельных операций.

По умолчанию `max_parallel_workers = 8`.

Значение данного параметра, которое превышает значение `max_worker_processes`, не будет действовать, так как параллельные рабочие процессы берутся из пула рабочих процессов, ограничиваемого этим параметром.

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

```
backend_flush_after (integer)
```

Если при одном обслуживающем процессе записывается больше чем backend_flush_after байт, сервер дает указание ОС произвести запись этих данных в нижележащее хранилище. Это ограничивает объем «грязных» данных в страничном кэше ядра и уменьшает вероятность затормаживания при выполнении fsync в конце контрольной точки или, когда ОС сбрасывает данные на диск большими порциями в фоне.

По умолчанию backend_flush_after = 0, процесс отключен.

Параметр действует не на всех платформах и может принимать значение от 0 (управление отложенной записью отключается) до 2 Мб. Если BLCKSZ отличен от 8 Кб, максимальное значение корректируется пропорционально.

```
old_snapshot_threshold (integer)
```

Задаёт минимальное время, которое позволяет использовать снимок без ошибки о давности снимка. Данный параметр можно задать только при запуске сервера.

По умолчанию old_snapshot_threshold = -1 отключает этот процесс. Нужные значения для производственной среды могут лежать в интервале от нескольких часов до нескольких дней. Заданное значение округляется до минут. При многих видах нагрузки критичное замусоривание базы или зацикливание идентификаторов транзакций может происходить за меньший промежуток времени.

4.7. Параметры межсетевого взаимодействия

СУБД «Jatoba» имеет клиент-серверную архитектуру. Для подключения к СУБД используются протоколы Libpq и Jadog.

Libpq – протокол, который используется для подключения к БД пользователей. Протокол Libpq реализован в виде драйвера «Driver Libpq» и обязательно требуется для работы приложений с СУБД. Дополнительно можно использовать ODBC драйвера, если приложения поддерживают подключение к СУБД через API ODBC.

Jadog – проприетарный протокол, используется для подключения к СУБД привилегированных пользователей, который обеспечивает взаимодействие между СВТ и

сервером СУБД в среде функционирования изделия. При этом не используется драйвер протокола Libpq. Инициирование подключения осуществляет клиентское приложение.

Протоколы Libpq и Jadog используют стек протоколов TCP/IP и Unix-сокеты в клиент-серверном исполнении Изделия.

Параметры стека протоколов приведены в таблице 4.1.

Таблица 4.1 – Параметры протоколов используемых СУБД

Компонент	Наименование протокола	Протокол	Порты
СУБД	Database port (db_port)	Libpq	5432
	Аутентификации LDAP	LDAP	389 (AD, ALD Pro, FreeIPA) по умолчанию
		LDAPS	636 (SAMBA) по умолчанию
	Аутентификации GSSAPI	NTLM/Kerberos	88, 445
	Аутентификации SSPI	NTLM/Kerberos	88, 445
	Аутентификации Radius	Radius	1812, 1813
	Аутентификации TLS/SSL (сертификаты)	TLS/SSL	5432
jaDog	Jadog TCP port (user_interface)	TCP	54321, 54322
	Jadog PORT number (port)	Jadog	12345, (Custom)
	Jadog searching protocol port (jadog_search_port)	Jadog	12346
	REST API	REST API	54443
ja_Hipe_Cluster	Database port (db_port)	Libpq	5432
	Протокол аутентификации SSL	SSL	5432, 443
ja_Inventory	Database port (db_port)	Libpq	5432
ja_Log	Database port (db_port)	Libpq	5432
	Протокол аутентификации SSL	SSL	443, 10051
ja_Sync_Ldap	Протокол аутентификации LDAP	LDAP	389 (AD, ALD Pro, FreeIPA) по умолчанию
		LDAPS	636 (SAMBA)

Компонент	Наименование протокола	Протокол	Порты
			по умолчанию
JDS	Database port (db_port)	Libpq	5432
	Протокол передачи данных HTTPS	HTTPS	443, 5000
	Протокол передачи данных HTTP	HTTP	9000
	Протокол электронной почты	SMTP	25, 587
	Протокол аутентификации SSL	SSL	464
	Протокол передачи сообщений на веб-сервер ZULIP	ZULIP	443
	Протокол удалённого управления операционной системой	SSH	22
	Jadog PORT number (port)	Jadog	12345, (Custom)
	Протокол аутентификации LDAP	LDAP	389 (AD, ALD Pro, FreeIPA), 636 (SAMBA)
	Database port (db_port)	Libpq	5433
Prometheus	Протокол передачи данных HTTP	HTTP	9090
	Протокол удалённого управления операционной системой	SSH	22
Alert manager	Протокол передачи данных HTTP	HTTPS	9093
	Протокол удалённого управления операционной системой	SSH	22
	Протокол электронной почты	SMTP	25
node_exporter	Протокол передачи данных HTTP	HTTP	9100
postgres_exporter	Протокол передачи данных HTTP	HTTP	9187
	Database port (db_port)	Libpq	5432
sql_exporter	Протокол передачи данных HTTP	HTTP	9399
	Database port (db_port)	Libpq	5432
pg_ProBackup	Database port (db_port)	Libpq	5433
	Протокол удалённого управления операционной системой	SSH	22
pgSQL-HTTP	Протокол передачи данных HTTP	HTTP	80, 443, Custom proxy port

Клиентское приложение образует с серверной частью СУБД канал (сессию) взаимодействия по специальным протоколам, основанным на сообщениях.

Пользователи БД используют протокол Librq и порт 5432.

Администратор СУБД и администраторы БД используют протокол Jadog. Подключение к СУБД может происходить по портам 54321 и 54322. В качестве резервного интерфейса подключения может использоваться подключение по протоколу Librq на порт 5432.

Схема подключения представлена на рисунке 4.1.

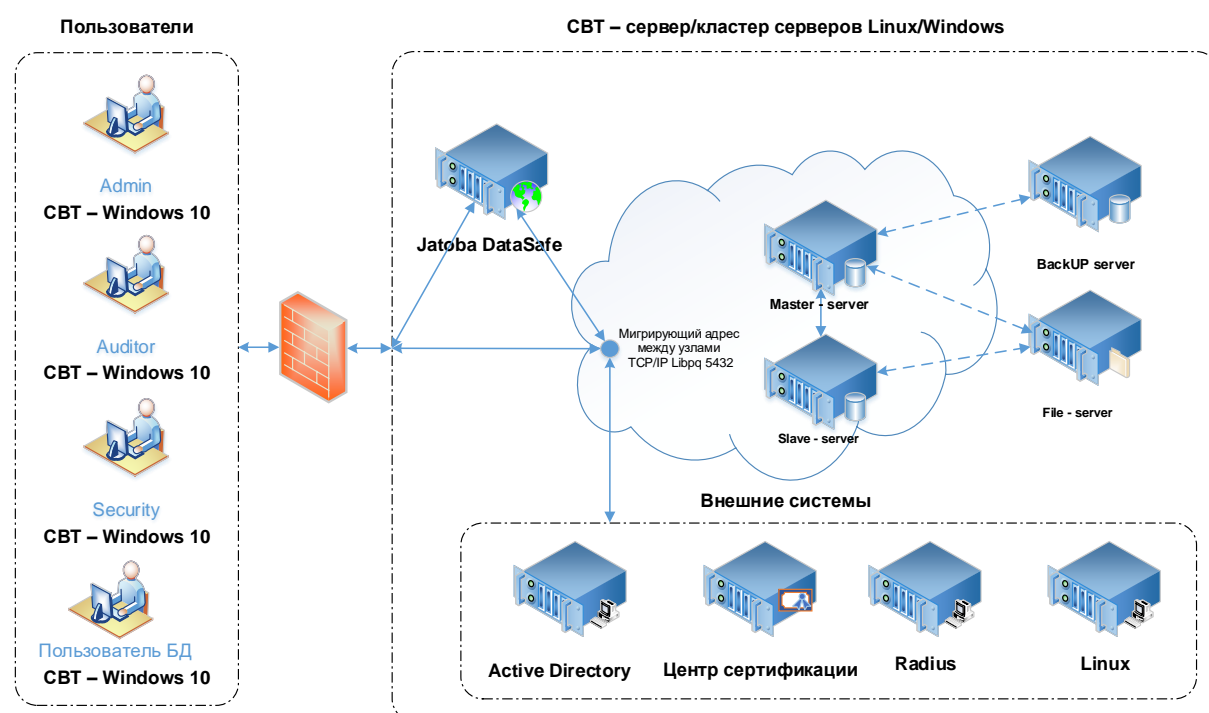


Рисунок 4.1 – Интерфейсы подключения СУБД

Протокол HTTPS по порту 443 используется для взаимодействия компонента «Jatoba Data Safe» с СУБД и служебной СУБД.

4.7.1. Основные параметры, связанные с настройкой сети

```
listen_addresses (string)
```

Указывает TCP/IP-адреса, по которым сервер будет принимать подключения клиентских приложений. Значение принимает форму списка имен и/или числовых IP-адресов компьютеров, разделенную запятыми. Специальный элемент * обозначает все

доступные IP-интерфейсы. Запись 0.0.0.0 позволяет задействовать все адреса IPv4, а :: – позволяет задействовать все адреса IPv6. Если список пуст, сервер не будет привязываться ни к какому IP-интерфейсу, в этом случае подключиться к нему можно будет только через Unix-сокеты.

По умолчанию – localhost, что позволяет устанавливать подключение к серверу по TCP/IP только через локальный интерфейс «замыкания». Параметр можно задать только при запуске сервера.

```
port (integer)
```

TCP-порт, запускаемый сервером.

По умолчанию – 5432.

Порт используется для всех IP-адресов, через которые сервер принимает подключения. Параметр можно задать только при запуске сервера.

```
max_connections (integer)
```

Определяет максимальное количество одновременных подключений к серверу БД.

По умолчанию – 100 подключений. Это число может быть меньше, если настройки ядра накладывают ограничения (определяется в процессе initdb). Параметр можно задать только при запуске сервера.

Максимальное значение количество одновременных подключений к серверу БД составляет - 262 143 подключений.

Для ведомого сервера значение параметра должно быть больше или равно значению на ведущем. В обратном случае запросы на ведомом сервере не будут разрешены.

```
superuser_reserved_connections (integer)
```

Определяет количество «слотов» подключений, зарезервированных для соединений суперпользователями. Одновременно могут быть активны не более max_connections подключений. Когда число активных одновременных подключений \geq max_connections – superuser_reserved_connections, новые подключения принимаются

только для суперпользователей, все остальные подключения, в том числе подключения для репликации, запрещаются.

По умолчанию резервируются три соединения. Это значение должно быть `< max_connections –max_wal_senders`. Задать этот параметр можно только при запуске сервера.

```
unix_socket_directories (string)
```

Задаёт каталог Unix-сокета, через который сервер будет принимать подключения клиентских приложений. Можно создать несколько сокетов, перечислив в этом значении несколько каталогов через запятую. Пробелы между записями игнорируются, если в пути каталога содержатся пробелы, его нужно прописать в двойных кавычках. При пустом значении сервер не будет работать с Unix-сокетами, в этом случае к нему можно подключиться только по TCP/IP.

Значение по умолчанию обычно `/tmp`. Его можно изменить во время сборки. Задать параметр можно только при запуске сервера.

В дополнение к самому файлу сокета, который называется `.s.PGSQL.nnnn` (где `nnnn` – номер порта сервера), в каждом каталоге `unix_socket_directories` будет создан обычный файл с именем `.s.PGSQL.nnnn.lock`. Ни один из файлов нельзя удалять вручную.

Этот параметр не действует в ОС Windows, так как в ней нет Unix-сокетов.

4.8. Настройка производительности СУБД «Jatoba»

```
std_fuzz_factor(numeric)
```

Глобальный параметр `«std_fuzz_factor»` позволяет расширить диапазон оценок планов при сравнении различных планов-кандидатов и таким образом, включить больше планов в кандидаты на выяснение лучшего.

Чем больше значение параметра `«std_fuzz_factor»`, тем больше кандидатов будет рассматривать планировщик запросов. В результате в список кандидатов могут быть включены более оптимальные планы с потенциально большей производительностью, но схожие по оценке с другими кандидатами. Таким образом, регулируя значение параметра может повышаться общая производительность СУБД и скорость исполнения запросов.

Значением по умолчанию параметра является – «1,01». Возможный диапазон значений параметра «std_fuzz_factor» варьируется от 0,9 до 1,9.

Увеличение значение параметра приводит к возрастанию временных затрат на формирование планов запросов. Эффективное значение параметра определяется практической эксплуатацией в зависимости от сложности выполняемых запросов в СУБД.

4.9. Поддержка возможности включения принудительной очистки высвобождаемых блоков в файлах данных (Data wiping/Zeroing)

СУБД «Jatoba», используя средства сертифицированных ОС, приведенные в таблице 1.1, обеспечивает удаление:

- баз данных и журналов;
- объектов доступа базы данных;

используемых СУБД, путем:

- многократной перезаписи уничтожаемых (стираемых) объектов файловой системы специальными битовыми последовательностями;
- перезаписи модифицированных участков объектов файловой системы при выполнении операции удаления или в отложенном режиме через промежуток времени.

Что соответствует документу «Требования по безопасности информации к системам управления базами данных (выписка)», утвержденному приказом ФСТЭК России от 14.04.2023 № 64, в части требований к очистке памяти в СУБД.

Для реализации данного механизма требуется воспользоваться документацией на используемую ОС.

5. ОСНОВНЫЕ ОПЕРАЦИИ В СУБД «ЯТОВА»

5.1. Создание ролей в БД

Роли баз данных являются глобальными для всей СУБД, не для отдельной БД.

При начальной установке СУБД содержит одну предопределенную роль postgres, обладающую максимальными привилегиями (SUPERUSER).



Для создания других ролей нужно подключиться под ролью postgres.

Создать роль возможно двумя способами:

- SQL-командой:

```
CREATE ROLE <имя>;
```

- утилитой командной строки:

```
createuser <имя>
```

Подключение к серверу БД выполняется от имени и с правами конкретной учетной записи с указанием конкретной БД.

Наличие доступа к объектам СУБД и БД, а также возможность выполнения команд, определяется назначенными роли атрибутами и системными привилегиями.

5.2. Удаление ролей в БД

Роль может быть удалена двумя способами:

- SQL-командой:

```
DROP ROLE <имя>;
```

- утилитой командной строки:

```
dropuser <имя>
```

Роли могут владеть объектами БД и иметь права доступа к объектам других пользователей. При удалении роли необходимо убедиться, что объекты, принадлежащие данной роли, были переданы другой роли или удалены.

Право владения объектами можно передавать в индивидуальном порядке с использованием команды ALTER. Например, для таблиц команда ALTER выглядит следующим образом:

```
ALTER TABLE <имя_таблицы> OWNER TO <принимающая_роль>;
```

Переназначение владения для отдельных объектов может быть проблематичным, если в БД пользователя насчитывается большое количество объектов. Для переназначения права владения всеми объектами с удаляемой роли можно воспользоваться командой REASSIGN OWNED.

```
REASSIGN OWNED BY <удаляемая_роль> TO <новая_роль>;
```

При удалении объектов, которыми владеет удаляемая роль, можно использовать следующие команды.

Для удаления отдельных объектов БД используется команда DROP. Например, для удаления отдельной таблицы:

```
DROP TABLE <имя_таблицы>;
```

Для массового удаления объектов, принадлежащих удаляемой роли, используется команда DROP OWNED.

```
DROP OWNED BY <удаляемая_роль>;
```



DROP OWNED не удаляет табличные пространства и базы данных целиком. Это необходимо сделать вручную, убедившись, что удаляемые данные не представляют ценности.

5.3. Создание БД

Для создания базы данных сервер СУБД «Jatoba» должен быть развернут и запущен.



Пользователь, создающий БД, автоматически назначается ее владельцем. Владелец может удалить свою базу, что приведет к удалению всех объектов. Только пользователь, обладающий привилегиями CREATEDB и SUPERUSER, может создавать новые БД.

Создать БД можно двумя способами:

- SQL-командой:

```
CREATE DATABASE <имя>;
```

Для выполнения команды CREATE DATABASE необходимо подключение к серверу баз данных. При установке СУБД всегда содержит служебную БД postgres, к которой необходимо подключиться для создания других БД.

- Утилитой командной строки createdb. Если имя БД не указано в параметрах командной строки, то эта утилита создаст базу данных с именем текущего пользователя:

```
createdb <имя>
```

В случае создания базы данных одним пользователем для другого пользователя и назначении его владельцем используется одна из следующих команд:

- SQL-команда:

```
CREATE DATABASE <имя_базы> OWNER <имя_роли>;
```

- Утилита командной строки:

```
createdb -O <имя_роли> <имя_базы>
```

5.4. Удаление БД

Удалить базу данных могут только или владелец БД, или пользователь, обладающий максимальными привилегиями (SUPERUSER).

Удалить БД можно двумя способами:

- SQL-командой:

```
DROP DATABASE <имя>;
```

– Утилитой командной строки

```
dropdb <имя>
```

Выполнить команду удаления БД невозможно, пока существует хоть одно подключение к базе.



При удалении БД удаляются все ее объекты. Удаление БД необратимая операция.

5.5. Создание внешнего ТП



Пользователь, создающий внешнее ТП, автоматически назначается ее владельцем. Владелец ТП может удалить свое внешнее ТП. Только пользователь, обладающий привилегиями CREATE TABLESPACE и SUPERUSER, может создавать новые внешние ТП (см. п.п. 6.2.3).

Создать внешнее табличное пространство (ТП) можно с помощью следующей команды:

```
CREATE TABLESPACE 'name_tablespace' LOCATION '/path';
```

Где /path – путь к каталогу, в котором будет располагаться внешнее табличное пространство. Если указанный путь не существует внешнее ТП не будет создано.

Пример:

```
CREATE TABLESPACE 'tablespace1_db1' LOCATION  
'/home/user_db1/tablespace1';
```



Пользователь ОС postgres должен иметь доступ на чтение/запись в указываемый каталог. Если у пользователя ОС postgres нет доступа к указываемому каталогу, то внешнее ТП не будет создано.



Каталог, в котором должно располагаться внешнее ТП не должен содержать других внешних ТП. В этом случае при выполнении команды выше внешнее ТП не будет создано.

После создания каталога внешнего ТП создается символическая ссылка с именем OID ТП в каталоге /var/lib/jatoba/6/data/pg_tblspc.

Для резервного узла кластера компонента «jaDog», куда копируется внешнее табличное пространство (ТП) с главного узла, указанный каталог не должен содержать внешних ТП. Например, если на главном узле внешнее ТП создано в каталоге /home/user_db1/tablespace1, то на резервном узле(ах) этот каталог должен быть пустым — соответствующее ТП там присутствовать не должно. При попытке копирования с главного узла создаваемое внешнее ТП в существующий каталог с другим внешним ТП на резервном узле возможно завершение работы сервисов компонента «jaDog». После устранения ограничения необходимо перезапустить сервисы компонента «jaDog» при помощи команды:

```
systemctl restart jadog
```



Не рекомендуется создавать вложенные друг в друга ТП – так как это может привести к ошибкам в работе служебных утилит pg_rewind, pg_basebackup и других. Например, при выполнении синхронизации pg_rewind возможно возникновение ошибок порядка создания каталогов ТП.

5.6. Удаление внешнего ТП

Удалить внешнее ТП могут только или его владелец, или пользователь, обладающий максимальными привилегиями (SUPERUSER).



Перед удалением внешнего ТП из него необходимо удалить объекты БД, относящейся к данному ТП.

Удалить внешнее табличное пространство (ТП) можно с помощью следующей команды:

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

```
DROPE TABLESPACE 'name_tablespace';
```

Пример:

```
DROPE TABLESPACE 'tablespace1_db1';
```

При удалении каталога внешнего ТП удаляется символическая ссылка с именем OID ТП из каталога /var/lib/jatoba/6/data/pg_tblspc.

Процедура удаления внешнего ТП в обязательном порядке записывается в журналы информационной и системной безопасности.

В случае резервного узла кластера компонента «jaDog», на который выполнялось реплицирование внешнего ТП с главного узла, после удаления ТП и в процессе последующей синхронизации каталог ТП также будет удален.

6. НАСТРОЙКА БЕЗОПАСНОСТИ СУБД «ЈАТОВА»

Ролевая модель СУБД должна соответствовать двум основным принципам:

- Разумной достаточности;
- Назначению минимально необходимых прав и привилегий пользователям, администраторам и лицам, обеспечивающим функционирование (УПД.5).

Контроль за выданными правами в СУБД доступно выполнять в компоненте «Jatoba data safe», в разделах:

- Раздел «Матрица доступа» (Access matrix);
- Раздел «Анализ рисков» (User Risk).

Как описано в документе, Руководство по настройке. Часть 7. Пользовательский веб-интерфейс для администраторов. Компонент «Jatoba data safe» 643.72410666.00067-07 98 01-07.

6.1. Идентификация и аутентификация субъектов доступа

Идентификация пользователей в СУБД «Jatoba» осуществляется по уникальным именам. Для создания пользователей в СУБД «Jatoba» необходимо от учетной записи администратора СУБД выполнить следующую команду:

```
CREATE ROLE <имя учетной записи пользователя>;
```

Аутентификация пользователей осуществляется с использованием паролей (метод аутентификации (md5, password)).

Для настройки аутентификации пользователей необходимо:

- настроить конфигурационный файл pg_hba.conf под необходимый метод аутентификации;
- настроить парольную политику.

6.1.1. Настройка конфигурационного файла pg_hba.conf

Для настройки конфигурационного файла pg_hba.conf необходимо выполнить следующие действия:

1) От учетной записи администратора СУБД подключиться к ОС и открыть файл pg_hba.conf¹⁾.

2) В открывшемся файле pg_hba.conf внести необходимые записи в определенном формате:

TYPE	DATABASE	USER	ADDRESS	METHOD
<i>local</i>	<i>база</i>	<i>пользователь</i>	————	<i>метод-аутентификации</i>
<i>host</i>	<i>база</i>	<i>пользователь</i>	<i>адрес</i>	<i>метод-аутентификации</i>
<i>hostssl</i>	<i>база</i>	<i>пользователь</i>	<i>адрес</i>	<i>метод-аутентификации</i>
<i>hostnossl</i>	<i>база</i>	<i>пользователь</i>	<i>адрес</i>	<i>метод-аутентификации</i>
<i>host</i>	<i>база</i>	<i>пользователь</i>	<i>IP-адрес</i> <i>маска</i>	<i>метод-аутентификации</i>
<i>hostssl</i>	<i>база</i>	<i>пользователь</i>	<i>IP-адрес</i> <i>маска</i>	<i>метод-аутентификации</i>
<i>hostnossl</i>	<i>база</i>	<i>пользователь</i>	<i>IP-адрес</i> <i>маска</i>	<i>метод-аутентификации</i>

Значение полей из файла pg_hba.conf представлены в Приложении 1.

3) Перезагрузить сервис СУБД «Jatoba» в:

– ОС Windows Server при помощи команд:

```
net stop JatobaServer;
net start JatobaServer
```

– ОС семейства GNU/Linux при помощи команды:

```
systemctl restart jatoba-<ver>
```

¹⁾ Местонахождение файла pg_hba.conf по умолчанию:

– ОС семейства GNU/Linux: /var/lib/jatoba/<ver>/data/pg_hba.conf;

– ОС Windows: C:\Program Files\GIS\Jatoba\<ver>\data\pg_hba.conf.

6.1.2. Настройка SSL

СУБД «Jatoba» имеет возможность использования защищённого сетевого трафика и аутентификацию по SSL сертификату. Аутентификация клиента по SSL сертификату позволяет серверу проверить личность подключающегося, подтверждая, что сертификат X.509, представленный клиентом, подписан центром сертификации. Рекомендуется использовать только доверенные центры сертификации для выдачи сертификатов клиенту и серверу. Для установки SSL необходимо установить следующие конфигурационные параметры файла postgresql.conf:

```
ssl = on
ssl_cert_file = '/usr/jatoba-5/etc/jatoba/server.crt'
ssl_key_file = '/usr/jatoba-5/etc/jatoba/server.key'
ssl_ca_file = '/usr/jatoba-5/etc/jatoba/root.crt'
```

При выпуске серверного сертификата, поле ASN, а при его отсутствии CN, должно соответствовать доменному имени сервера или его IP-адресу. В Unix-подобных системах к файлу server.key должен быть запрещён любой доступ группы и всех остальных, чтобы установить такое ограничение, выполнить:

```
chmod 0600 server.key
```

Файл pg_hba.conf для входящих IP-адресов должен содержать следующие строки:

#	TYPE	DATABASE	USER	ADDRESS	METHOD	
hostssl	all		all	<ip6/mask>	cert	clientcert=verify-full
hostssl	all		all	<ip/mask>	cert	clientcert=verify-full

Также с помощью файла pg_hba.conf изменяя значения колонок DATABASE и USER можно ограничить доступ до конкретной роли пользователя или БД.

Самая строгая степень проверки SSL сертификата: clientcert=verify-full. При этом типе авторизации проверяется соответствие значения поля CN пользовательского сертификата имени пользователя СУБД.

Если установлен параметр ssl_crl_file или ssl_crl_dir, также проверяются списки отзыва сертификатов.

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

Степень проверки `clientcert=verify`-са указывает на проверку только подлинности входящего клиентского сертификата. А степень проверки `clientcert=require` указывает только на создание SSL соединения, без проверок подлинности X.509 сертификатов.

OpenSSL предоставляет набор шифров и алгоритмов аутентификации разной защищённости. Список шифров может быть задан в файле конфигурации OpenSSL. Также можно задать конкретные используемые в БД шифры, указав их в параметре `ssl_ciphers` в `postgresql.conf`.

Если в конфигурационном файле `postgresql.conf` установлен параметр `ssl_crl_file` или `ssl_crl_dir`, то X.509 сертификат проверяется в списках отзыва сертификатов: (Certificate Revocation List, CRL).

6.1.3. Настройка парольной политики. Компонент «SecurityProfile»

Парольная политика – это набор правил и требований, предназначенных для повышения безопасности СУБД «Jatoba» путем принуждения использования пользователями сложных и безопасных паролей.

В СУБД «Jatoba» парольные политики реализуются с использованием компонента SecurityProfile. Компонент SecurityProfile позволяет администраторам БД централизованно управлять значимыми аспектами паролей пользователей: от их сложности и сроков действия до блокировки учетных записей при нарушении правил.

В компоненте SecurityProfile применяются три основных метода:

- использование парольной политики по умолчанию;
- использование встроенных профилей;
- создание и настройка дополнительных профилей парольной политики.



Так как компонент SecurityProfile вносит ряд ограничений на использование паролей это может привести к блокировке существующих пользователей. Следует внимательно относиться к уже работающей системе, некоторые операции могут завершиться с ошибкой после установки расширения SecurityProfile.



Необходимо обратить внимание, что целесообразнее формировать парольную политику до создания новых учетных записей пользователей.



После выполнения операций `pg_dump/pg_dumpall` и `pg_restore` - требуется сменить пароли для пользователей СУБД, как описано в п. 6.2.4, SQL-командой:

```
ALTER ROLE <имя учетной записи пользователя> PASSWORD  
'<пароль пользователя>';
```

Установка компонента SecurityProfile в ОС описана в документе «Защищенная система управления базами данных «Jatoba». Руководство по установке».

Удаление компонента SecurityProfile из ОС описано в п.п. 6.1.4 данного руководства.

Использовать следующие функциональные возможности компонента SecurityProfile может привилегированный пользователь или пользователь имеющий право доступа к схеме «securityprofile»:

- смена пароля пользователя;
- снятие блокировки пользователя;
- создание профиля парольной политики;
- назначение пользователю профиля парольной политики;
- смена параметров профиля парольной политики и т.д.

Использовать следующие функциональные возможности компонента SecurityProfile может непривилегированный пользователь:

- смена пароля пользователя.

6.1.3.1 Установка расширения securityprofile в СУБД

Для расширения securityprofile в СУБД «Jatoba» необходимо выполнить следующие действия:

1) От учетной записи администратора СУБД подключиться к ОС и открыть файл postgresql.conf²⁾.

2) В файле postgresql.conf прописать в параметр shared_preload_libraries значение securityprofile:

```
shared_preload_libraries = 'securityprofile'
```



Указание в конфигурационном файле postgresql.conf опции shared_preload_libraries = 'securityprofile' активирует компонент управления парольными политиками и создается политика по умолчанию.

3) В файле postgresql.conf прописать следующую строку:

```
securityprofile.db_name = 'dbname'
```

Параметр «dbname» определяет имя БД, в которой будет создаваться или уже создано расширение securityprofile, значение по умолчанию postgres.



ВАЖНО! При совместной работе SecurityProfile в одной СУБД с компонентами «jaPooler», «jaDog» и/или «ja_Log» в параметре securityprofile.db_name необходимо указывать наименование БД, в которую установлено расширение этих компонентов.

4) После этого необходимо выполнить перезапуск сервиса СУБД средствами ОС;

5) Подключиться к БД, указанной в параметре securityprofile.db_name и выполнить установку расширения securityprofile:

```
CREATE EXTENSION securityprofile;
```



В случае установки расширения в БД, которая отсутствует в параметре securityprofile.db_name пользователю выводится сообщение:

Securityprofile should be installed into "<dbname>" database

²⁾ Местонахождение файла postgresql.conf по умолчанию в:

- ОС семейства GNU/Linux: /var/lib/jatoba/<ver>/data/postgresql.conf;
- ОС Windows Server: C:\Program Files\GIS\Jatoba\<ver>\data\postgresql.conf.

6.1.3.2 Параметры парольной политики по умолчанию

Парольная политика по умолчанию активируется после установки расширения SecurityProfile.



После установки расширения SecurityProfile для всех существующих пользователей необходимо выполнить повторную установку паролей.

Параметры парольной политика по умолчанию настраивается при помощи GUC и указаны в таблице 6.1.

Команды управления профилем парольной политика в СУБД «Jatoba» представлены в таблице 6.11.

Таблица 6.1 - Параметры профиля парольной политики по умолчанию

Параметр	Примечание	Параметры в профиле по умолчанию	Мин-е значение	Макс-е значение
Параметры для конфигурирования парольной политики по умолчанию				
securityprofile.default_profile	имя профиля по умолчанию	'default'		
securityprofile.special_chars	пароль должен содержать указанные символы (набор символов не является обязательным и может быть изменен)	\!"#\$%&()*+,-./:;<=>?@[]^_`{ }~		
securityprofile.lower_case_count	пароль должен содержать как минимум 1 символ в нижнем регистре	1	0	256
securityprofile.upper_case_count	пароль должен содержать как минимум 1 символ в верхнем регистре	1	0	256
securityprofile.numbers_count	пароль должен содержать как минимум 1 цифру	1	0	256
securityprofile.special_count	пароль должен содержать как минимум 1 спец. символ из заданного набора special_chars	1	0	256
securityprofile.minimum_length	минимальная длина пароля равна 6 символам	6	6	256
securityprofile.maximum_length	максимальная длина пароля равна 32 символам	32	6	256
securityprofile.minimum_changes	минимальное количество изменений, которое должен содержать новый пароль по сравнению с предыдущим	0	0	256
securityprofile.failed_login_attempts	количество неудачных попыток входа в СУБД	10	0	Int_Max
securityprofile.password_lock_time	время, на которое блокируется пользователь в СУБД (1 час в секундах) Значение «-1» означает, что не будет происходить блокировка пользователя на заданное время, т.е. отсутствие блокировки. В противном случае при нарушении политики	3600	-1	Int_Max

Параметр	Примечание	Параметры в профиле по умолчанию	Мин-е значение	Макс-е значение
	профиля - пользователь блокируется на указанное в параметре время.			
securityprofile.failed_login_attempts_max_time_sec	время в течении которого допустимо ошибиться	0	0	Int_Max
securityprofile.password_min_life_time	время в секундах, в течение которого пароль должен использоваться и не может быть изменен	0	0	Int_Max
securityprofile.password_life_time	время в секундах, в течение которого может быть использован текущий пароль (180 дней в секундах)	15 552 000	-1	Int_Max
securityprofile.password_grace_time	время в секундах, в течение которого пользователь может использовать текущий пароль с напоминанием о необходимости его сменить до блокировки аккаунта. Время прибавляется к времени, установленному в securityprofile.password_life_time	0	-1	Int_Max
securityprofile.password_reuse_time	время между повторным использованием одного и того же пароля (в секундах)	0	-1	Int_Max
securityprofile.password_reuse_max	повторное использование пароля запрещено (Для повторного использования пароля без ограничений надо указать password_reuse_time=-1 и password_reuse_max = -1)	-1	-1	Int_Max
securityprofile.store_password_encrypted	хранение паролей в закрытом виде	True	True	False
securityprofile.user_idle_days_max	Политика компонента, контролирующая, что пользователь должен быть заблокирован, если период его неактивности (отсутствие входов в СУБД) превысил заданное количество дней	45	-1	Int_Max
Параметры для конфигурирования размера кэша расширения				

Параметр	Примечание	Параметры в профиле по умолчанию	Мин-е значение	Макс-е значение
securityprofile.profiles_cache_limit	максимальное количество профилей, хранимых в кэше	10	0	Int_Max
securityprofile.accounts_cache_limit	максимальное количество пользовательских аккаунтов, хранимых в кэше	1000	0	Int_Max
securityprofile.password_history_cache_limit	максимальное количество парольных хэшей (md5) хранимых в кэше	10000	0	Int_Max
securityprofile.status_cache_limit	максимальное количество записей о статусах блокировок всех пользователей	10000	0	Int_Max

В таблице указываемые параметры после знака «=» не являются абсолютными. Они могут быть изменены в диапазоне от минимального до максимального значения в соответствии с требованиями парольной политики. Значение «INT_MAX» обозначает числовое значение для переменной «INT» равным значению 2 147 483 647.

Значение «-1» означает запрет или блокировку.

6.1.3.3 Создание нового профиля парольной политики

- 1) В файле postgresql.conf прописать следующую строку:

```
securityprofile.default_profile = 'profile_name'
```

где profile_name – название создаваемого профиля.

Или при помощи SQL-команды:

```
ALTER SYSTEM SET securityprofile.default_profile =  
'profile_name';
```



Указываемый в параметре securityprofile.default_profile профиль парольной политики будет использоваться по умолчанию для всех новых пользователей.

Если параметр securityprofile.default_profile не указан, то к новым пользователям будет применяться профиль default. Если профиль не создан, но указан в параметре securityprofile.default_profile, то новым пользователям он не применяется и применяется профиль default.

В качестве профиля по умолчанию может быть выбран один из встроенных профилей парольных политик (см. п.п. 6.1.3.8.1-6.1.3.8.6).

- 2) После этого необходимо выполнить чтение файла конфигурации СУБД:

```
SELECT pg_reload_conf();
```

- 3) Создать в СУБД новый профиль с помощью SQL-команды:

```
SELECT securityprofile.create_profile('profile_name');
```

- 4) После создания профиля, для новых пользователей СУБД, он будет применяться назначаться автоматически. Параметра этого профиля изменяются с помощью соответствующих команд, указанных в таблице 6.10.

Чтобы посмотреть, какие профили назначены пользователям, надо выполнить запрос

```
SELECT  
(SELECT
```

```
rolname FROM pg_authid WHERE oid=accroleoid) rolename,  
case WHEN accprofileoid=0  
THEN 'builtin default'  
ELSE (select prflprofilename FROM securityprofile.profiles  
WHERE prflprofileoid=accprofileoid) END  
profilename  
FROM securityprofile.accounts;
```

6.1.3.4 Отключение пользовательского профиля парольной политики

Чтобы отключить созданный пользовательский профиль (см. п. 6.1.3.3) и переключиться на профиль по умолчанию необходимо удалить (или закомментировать) параметр `securityprofile.default_profile` в конфигурационном файле `postgresql.conf`.

При переключении на профиль по умолчанию требования к парольной политике существующих пользователей не изменяются, используются только для новых пользователей СУБД.

6.1.3.5 Смена профиля парольной политики для пользователя

Профиль можно сменить пользователю с помощью SQL-команды следующего синтаксиса:

```
SELECT securityprofile.bind_profile(['user'], ['profile_name']);
```

Пример:

```
SELECT securityprofile.bind_profile('manager', 'profile_hard');
```

6.1.3.6 Отдельные параметры компонента

securityprofile.user_idle_days_max

`user_idle_days_max` – это политика компонента, контролирующая, что пользователь должен быть заблокирован, если период его неактивности (отсутствие заходов в СУБД) превысил заданное количество дней.

Политика выполняет Требования к усилению УПД.1 (36) в части требований:

3) в информационной системе должно осуществляться автоматическое блокирование неактивных (неиспользуемых) учетных записей пользователей после периода времени неиспользования:

б) более 45 дней;

Управление значением параметром возможно через конфигурационный файл «postgresql.conf» и через одноименную функцию компонента от имени и с правами привилегированного пользователя. Значение устанавливается целыми числами, в днях.

Значение по умолчанию - 45 дней.

Минимально допустимое значение для параметра = -1 (без ограничений)

Максимально допустимое = INT_MAX (~2 млрд. дней)

Функция select_profile_user_idle_days_max устанавливает значение этой политики для заданного профиля.

```
select_profile_user_idle_days_max('profile', кол-во_дней);
```

Служебная информация securityprofile размещается в служебных таблицах SP (например, securityprofile.account, securityprofile.profile и т.п.). Чтобы SP мог быстро принимать решения о блокировке пользователей, данная информация из служебных таблиц должна быть всегда загружена в оперативную память (в противном случае, придется каждый раз обращаться в таблицу, что приведет к значительному падению производительности и подключения пользователей к СУБД). Для размещения информации в оперативной памяти создаются специальные кэш-таблицы (для быстрого поиска и изменения).

Параметры для конфигурирования размера кэша расширения

К таким параметрам относятся:

- securityprofile.profiles_cache_limit;
- securityprofile.accounts_cache_limit;
- securityprofile.password_history_cache_limit;
- securityprofile.status_cache_limit.

Указанные параметры задают максимальное количества записей, которые могут размещаться в оперативной памяти.

`securityprofile.profiles_cache_limit` – максимальное количество профилей, хранимых в кэше. Значение по умолчанию 10, т.е. по умолчанию «securityprofile» может работать не более чем, с 10 профилями.

`securityprofile.accounts_cache_limit` - максимальное количество пользовательских аккаунтов, хранимых в кэше. Значение по умолчанию 1000, т.е. по умолчанию «securityprofile» может работать не более чем, с 1000 пользователями.

`securityprofile.password_history_cache_limit` – максимальное количество парольных хэшей (md5) хранимых в кэше. Значение по умолчанию 10000 - это максимально возможное количество записей в оперативной памяти по истории паролей всех УЗ всех пользователей.

`securityprofile.status_cache_limit` - максимальное количество записей о статусах блокировок всех пользователей.

Администратор СУБД (SUPERUSER) имеет права увеличить данные ограничения по своему усмотрению.

Если в процессе эксплуатации SP появляется сообщение о нехватке памяти для размещения данных «securityprofile», Администратору СУБД требуется увеличить указанные выше параметры до нужных значений в соответствии с количеством обслуживаемых пользователей, профилей и статусов блокировок. Значения этих параметров ограничены лишь объемом оперативной памяти сервера СУБД (максимальное значение INT_MAX).

securityprofile.wait_extension_delay

`securityprofile.wait_extension_delay` – это параметр проверки установки расширения в СУБД.

Расширение SP имеет две части исполнения.

Одна (внутренняя) часть интегрируется в СУБД через `shared_preload_libraries = 'securityprofile'` и предназначена для реализации внутренних механизмов компонента «securityprofile». Таких как, встраивание в процесс аутентификации пользователей и

проверка политики, перехват запросов управления ролями (CREATE/ALTER/DROP ROLE) и внесение изменение в профили и УЗ и другие функции.

Вторая часть (SQL-часть) интегрируется в СУБД через установку расширения (CREATE EXTENSION securityprofile). В этой части идет создание служебной схемы «securityprofile», создание всех необходимых служебных таблиц и предопределенных профилей.

Работа обеих частей не может идти отдельно. Они должны быть установлены в СУБД совместно. Совместная установка разбита по времени:

— сначала Администратор СУБД устанавливается shared_preload_libraries и перезапускает СУБД:

— потом Администратор подключается к СУБД и устанавливает расширение SQL-командой: CREATE EXTENSION.

В промежутке времени между двумя этими действиями «securityprofile» должен проверять, установил ли Администратор расширение, и можно ли уже полноценно активировать свою работу.

В этой ситуации параметр wait_extension_delay задавал паузу во времени, через сколько «securityprofile» проверял, сделал ли уже Администратор установку расширения или нет.

Значение параметра по умолчанию 1с = 1000мс.

При повседневной работе наличие этого параметра приводила к большому количеству сообщений об ошибках, о том, что расширение «securityprofile» еще не установлено. Поэтому было принято решения, убрать этот параметр (версия «securityprofile»-2.1). Сейчас этот параметр объявлен "устаревшим", а именно можно задавать его значение в конф. файле, это не приведет к ошибке, но сам параметр более не действует. Все ожидания заменены на адаптивный алгоритм по логарифмической шкале. С каждым следующим циклом проверки таймауте увеличивается. Это не приводит к большому количеству сообщений в журнал аудита.

6.1.3.7 Обязательные действия, выполняемые сразу после установки расширения

Для учетной записи администратора СУБД в обязательном порядке, требуется задать новый пароль сразу после выполнения команды установки расширения.

В тех случаях, когда учетная запись пользователя в СУБД была создана по умолчанию до применения парольной политики, необходимо изменить пароль учетной записи при помощи команды:

```
ALTER ROLE <имя учетной записи пользователя> password '<пароль пользователя>';
```

После чего учетная запись пользователя привяжется к профилю «default».

Создание ролей пользователей при активированной парольной политике описано в п. 6.2.4.

6.1.3.8 Встроенные профили парольных политик

Компонент «securityprofile» имеет функциональную возможность распределять учетные записи по применяемым к ним парольным политикам. Как было описано ранее, возможно применять парольные политики по умолчанию, создавать собственные, либо использовать предварительно настроенные профили парольных политик.

К предварительно настроенным относятся:

- FSTEC_1_class – профиль для ИС первого класса защищенности (Таблица 6.2);
- FSTEC_2_class – профиль для ИС второго класса защищенности (Таблица 6.3);
- CIS – профиль, основанный на рекомендациях Center for Internet Security (Таблица 6.4);
- Corporate_1 – корпоративный профиль первого уровня для учетных записей пользователей (Таблица 6.5);
- Corporate_2 – корпоративный профиль второго уровня для учетных записей администраторов программных (программно-аппаратных средств) (Таблица 6.6);
- Corporate_3 – корпоративный профиль третьего уровня для технических (сервисных, служебных) учетных записей, используемых в технологических процессах ИС

или встроенных производителями программных (программно-аппаратных) средств в такие средства (Таблица 6.7).

Параметры в профиле установлены в зависимости от требований и могут быть изменены в сторону уменьшения до минимальных значений, как в профиле по умолчанию (default) (см. таблицу 6.1 так и в сторону увеличения до максимальных значений.

Максимальные значения обозначены параметром «INT_MAX». Это обозначение максимального значения для переменной «INT». При установке параметра максимального значения «INT_MAX» допускается числовое значение, которое меньше или равно 2 147 483 647.

Параметры в профилях парольных политик сформированы исходя из принципа разумной достаточности и установлены по минимальным значениям от требуемых либо усредненные.

Например

Рассмотрим реализацию части требований к усилению ИАФ.4 (1г) в соответствии с документом «Методический документ. Меры защиты информации в государственных информационных системах» (утв. ФСТЭК России 11.02.2014):

г) длина пароля не менее восьми символов, алфавит пароля не менее 70 символов, максимальное количество неуспешных попыток аутентификации (ввода неправильного пароля) до блокировки от 3 до 4 попыток, блокировка программно-технического средства или учетной записи пользователя в случае достижения установленного максимального количества неуспешных попыток аутентификации от 15 до 60 минут, смена паролей не более чем через 60 дней.

Длина пароля не менее восьми символов.

Требование выполняется параметром «securityprofile.minimum_length» со значением равным 8, что соответствует требованиям и менять его в меньшую сторону нельзя.

Максимальное количество неуспешных попыток аутентификации (ввода неправильного пароля) до блокировки от 3 до 4 попыток.

Требование выполняется параметром «securityprofile.failed_login_attempts» с значением равным 4, что соответствует максимальному значению и параметр может быть изменен, только до значения равному 3.

Блокировка учетной записи пользователя в случае достижения установленного максимального количества неуспешных попыток аутентификации от 15 до 60 минут.

Требование выполняется параметром «securityprofile.password_lock_time» с установленным значением 2700 сек, что равняется 45 минутам. Значение равно усредненному значению. В этом случае значение параметра «securityprofile.password_lock_time» могут быть изменены в диапазоне:

- от 15 минут (900 сек.);
- до 60 минут (3 600 сек.).

6.1.3.8.1 FSTEC_1_class

Профиль парольной политики «FSTEC_1_class» разработан в соответствии с требованиями к усилению ИАФ.4 (1г) в соответствии с документом «Методический документ. Меры защиты информации в государственных информационных системах» (утв. ФСТЭК России 11.02.2014) и может быть использован для ИС первого класса защищенности.

Таблица 6.2 – Параметры парольной политики FSTEC_1_Class по умолчанию

Параметр	Примечание	Параметры в профиле	Макс-е значение
Параметры для конфигурирования парольной политики			
securityprofile.profile_name	имя профиля по умолчанию	= ' FSTEC_1_Class '	
securityprofile.special_chars	пароль должен содержать указанные символы (набор символов не является обязательным и может быть изменен)	\!\"#\$%&()*+ , - ./:;<=>?@[]^ _`{ }~	
securityprofile.lower_case_count	пароль должен содержать как минимум 1 символ в нижнем регистре	1	256
securityprofile.upper_case_count	пароль должен содержать как минимум 1 символ в верхнем регистре	1	256

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

Параметр	Примечание	Параметры в профиле	Макс-е значение
securityprofile.numbers_count	пароль должен содержать как минимум 1 цифру	1	256
securityprofile.special_count	пароль должен содержать как минимум 1 спец. символ из заданного набора special_chars	1	256
securityprofile.minimum_length	минимальная длина пароля	8	256
securityprofile.maximum_length	максимальная длина пароля	256	256
securityprofile.minimum_changes	минимальное количество изменений, которое должен содержать новый пароль по сравнению с предыдущим	0	256
securityprofile.failed_login_attempts	количество неудачных попыток входа в СУБД	4	Int_Max
securityprofile.password_lock_time	время, на которое блокируется пользователь в СУБД (1 час в секундах) Значение «-1» означает, что не будет происходить блокировка пользователя на заданное время, т.е. отсутствие блокировки. В противном случае при нарушении политики профиля - пользователь блокируется на указанное в параметре время.	2700	Int_Max
securityprofile.failed_login_attempts_max_time_sec	время в течении которого допустимо ошибиться	300	Int_Max
securityprofile.password_min_life_time	время в секундах, в течение которого пароль должен использоваться и не может быть изменен	0	Int_Max
securityprofile.password_life_time	время в секундах, в течение которого может быть использован текущий пароль (в секундах)	5184000	Int_Max
securityprofile.password_grace_time	время в секундах, в течение которого пользователь может использовать текущий пароль с напоминанием о необходимости его сменить до блокировки аккаунта. Время прибавляется к времени,	0	Int_Max

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

Параметр	Примечание	Параметры в профиле	Макс-е значение
	установленному В securityprofile.password_life_time		
securityprofile.password_reuse_time	время между повторным использованием одного и того же пароля (в секундах)	0	Int_Max
securityprofile.password_reuse_max	повторное использование пароля	-1	Int_Max
securityprofile.store_password_encrypted	хранение паролей в закрытом виде	True	False

6.1.3.8.2 FSTEC_2_class

Профиль парольной политики «FSTEC_1_class» разработан в соответствии с требованиями к усилению ИАФ.4 (1в) в соответствии с документом «Методический документ. Меры защиты информации в государственных информационных системах» (утв. ФСТЭК России 11.02.2014) и может быть использован для ИС второго класса защищенности.

Таблица 6.3 – Параметры парольной политики FSTEC _2_Class по умолчанию

Параметр	Примечание	Параметры в профиле	Макс-е значение
Параметры для конфигурирования парольной политики			
securityprofile.profile_name	имя профиля по умолчанию	' FSTEC _2_Class '	
securityprofile.special_chars	пароль должен содержать указанные символы (набор символов не является обязательным и может быть изменен)	\!"#\$%&()*+ , - ./;<=>?@[]^ _`{ }~	
securityprofile.lower_case_count	пароль должен содержать как минимум 1 символ в нижнем регистре	1	256
securityprofile.upper_case_count	пароль должен содержать как минимум 1 символ в верхнем регистре	1	256
securityprofile.numbers_count1	пароль должен содержать как минимум 1 цифру	1	256

Параметр	Примечание	Параметры в профиле	Макс-е значение
securityprofile.special_count	пароль должен содержать как минимум 1 спец. символ из заданного набора special_chars	1	256
securityprofile.minimum_length	минимальная длина пароля	8	256
securityprofile.maximum_length	максимальная длина пароля	256	256
securityprofile.minimum_changes	минимальное количество изменений, которое должен содержать новый пароль по сравнению с предыдущим	0	256
securityprofile.failed_login_attempts	количество неудачных попыток входа в СУБД	5	Int_Max
securityprofile.password_lock_time	время, на которое блокируется пользователь в СУБД (1 час в секундах) Значение «-1» означает, что не будет происходить блокировка пользователя на заданное время, т.е. отсутствие блокировки. В противном случае при нарушении политики профиля - пользователь блокируется на указанное в параметре время.	1200	Int_Max
securityprofile.failed_login_attempts_max_time_sec	время в течении которого допустимо ошибиться	0	Int_Max
securityprofile.password_min_life_time	время в секундах, в течение которого пароль должен использоваться и не может быть изменен	0	Int_Max
securityprofile.password_life_time	время в секундах, в течение которого может быть использован текущий пароль	7776000	Int_Max
securityprofile.password_grace_time	время в секундах, в течение которого пользователь может использовать текущий пароль с напоминанием о необходимости его сменить до блокировки аккаунта. Время прибавляется к времени, установленному в securityprofile.password_life_time	0	Int_Max
securityprofile.password_reuse_time	время между повторным использованием одного и того же пароля (в секундах)	0	Int_Max

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

Параметр	Примечание	Параметры в профиле	Макс-е значение
securityprofile.password_reuse_max	повторное использование пароля	1	Int_Max
securityprofile.store_password_encrypted	хранение паролей в закрытом виде	True	False

6.1.3.8.3 CIS

Профиль парольной политики «CIS», разработан на основе рекомендаций Center for Internet Security – Центра интернет-безопасности, являющегося некоммерческой организацией, которая разрабатывает собственные контрольные показатели и рекомендации.

Таблица 6.4 – Параметры парольной политики «CIS» по умолчанию

Параметр	Примечание	Параметры в профиле	Макс-е значение
Параметры для конфигурирования парольной политики			
securityprofile.profile_name	имя профиля по умолчанию	'CIS'	
securityprofile.special_chars	пароль должен содержать указанные символы	\!"#\$%&()*+ , - ./:;<=>?@[]^ _`{ }~	
securityprofile.lower_case_count	пароль должен содержать как минимум 0 символов в нижнем регистре	0	256
securityprofile.upper_case_count	пароль должен содержать как минимум 0 символов в верхнем регистре	0	256
securityprofile.numbers_count	пароль должен содержать как минимум 1 цифру	1	256
securityprofile.special_count	пароль должен содержать как минимум 0 спец. символ из заданного набора special_chars	0	256
securityprofile.minimum_length	минимальная длина пароля	14	256
securityprofile.maximum_length	максимальная длина пароля	256	256
securityprofile.minimum_changes	минимальное количество изменений, которое должен содержать новый пароль по сравнению с предыдущим	0	256
securityprofile.failed_login_attempts	количество неудачных попыток входа в СУБД	5	Int_Max

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

Параметр	Примечание	Параметры в профиле	Макс-е значение
securityprofile.password_lock_time	время, на которое блокируется пользователь в СУБД (1 час в секундах) Значение «-1» означает, что не будет происходить блокировка пользователя на заданное время, т.е. отсутствие блокировки. В противном случае при нарушении политики профиля - пользователь блокируется на указанное в параметре время.	900	Int_Max
securityprofile.failed_login_attempts_max_time_sec	время в течении которого допустимо ошибиться	0	Int_Max
securityprofile.password_min_life_time	время в секундах, в течение которого пароль должен использоваться и не может быть изменен	6400	Int_Max
securityprofile.password_life_time	время в секундах, в течение которого может быть использован текущий пароль (в секундах)	7776000	Int_Max
securityprofile.password_grace_time	время в секундах, в течение которого пользователь может использовать текущий пароль с напоминанием о необходимости его сменить до блокировки аккаунта. Время прибавляется к времени, установленному в securityprofile.password_life_time	0	Int_Max
securityprofile.password_reuse_time	время между повторным использованием одного и того же пароля (в секундах)	0	Int_Max
securityprofile.password_reuse_max	повторное использование пароля	5	Int_Max
securityprofile.store_password_encrypted	хранение паролей в закрытом виде	True	False

6.1.3.8.4 Corporate_1

Профиль парольной политики «Corporate_1» – корпоративный профиль первого уровня для учетных записей пользователей.

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

Таблица 6.5 – Параметры парольной политики «Corporate_1»

Параметр	Примечание	Параметры в профиле	Макс-е значение
Параметры для конфигурирования парольной политики			
securityprofile.profile_name	имя профиля по умолчанию	'Corporate_1'	
securityprofile.special_chars	пароль должен содержать указанные символы	\!"#\$%&()*+ , - ./:;<=>?@[]^ _`{ }~	
securityprofile.lower_case_count	пароль должен содержать как минимум 1 символ в нижнем регистре	1	256
securityprofile.upper_case_count	пароль должен содержать как минимум 1 символ в верхнем регистре	1	256
securityprofile.numbers_count	пароль должен содержать как минимум 2 цифры	2	256
securityprofile.special_count	пароль должен содержать как минимум 1 спец. символ из заданного набора special_chars	1	256
securityprofile.minimum_length	минимальная длина пароля	6	256
securityprofile.maximum_length	максимальная длина пароля	256	256
securityprofile.minimum_changes	минимальное количество изменений, которое должен содержать новый пароль по сравнению с предыдущим	0	256
securityprofile.failed_login_attempts	количество неудачных попыток входа в СУБД	5	Int_Max
securityprofile.password_lock_time	время, на которое блокируется пользователь в СУБД (1 час в секундах) Значение «-1» означает, что не будет происходить блокировка пользователя на заданное время, т.е. отсутствие блокировки. В противном случае при нарушении политики профиля - пользователь блокируется на указанное в параметре время.	-1	Int_Max
securityprofile.failed_login_attempts_max_time_sec	время в течении которого допустимо ошибиться	0	Int_Max
securityprofile.password_min_life_time	время в секундах, в течение которого пароль должен	0	Int_Max
<div> <div>№ изменения: _____</div> <div>Подпись отв. лица: _____</div> <div>Дата внесения изм: _____</div> </div>			

Параметр	Примечание	Параметры в профиле	Макс-е значение
	использоваться и не может быть изменен		
securityprofile.password_life_time	время в секундах, в течение которого может быть использован текущий пароль (в секундах)	3888000	Int_Max
securityprofile.password_grace_time	время в секундах, в течение которого пользователь может использовать текущий пароль с напоминанием о необходимости его сменить до блокировки аккаунта.	0	Int_Max
securityprofile.password_reuse_time	время между повторным использованием одного и того же пароля (в секундах)	0	Int_Max
securityprofile.password_reuse_max	повторное использование пароля	5	Int_Max
securityprofile.store_password_encrypted	хранение паролей в закрытом виде	True	False

6.1.3.8.5 Corporate_2

Профиль парольной политики «Corporate_2» – корпоративный профиль второго уровня для учетных записей администраторов программных (программно-аппаратных средств).

Таблица 6.6 – Параметры парольной политики «Corporate_2»

Параметр	Примечание	Параметры в профиле	Макс-е значение
Параметры для конфигурирования парольной политики			
securityprofile.profile_name	имя профиля по умолчанию	'Corporate_2'	
securityprofile.special	пароль должен содержать указанные символы	\!"#\$%&()*+ , - ./:;<=>?@[^ _`{ }~	
securityprofile.lower_case_count	пароль должен содержать как минимум 1 символов в нижнем регистре	1	256
securityprofile.upper_case_count	пароль должен содержать как минимум 1 символ в верхнем регистре	1	256

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

Параметр	Примечание	Параметры в профиле	Макс-е значение
securityprofile.numbers_count	пароль должен содержать как минимум 2 цифры	2	256
securityprofile.special_count	пароль должен содержать как минимум 1 спец. символ из заданного набора special_chars	1	256
securityprofile.minimum_length	минимальная длина пароля	12	256
securityprofile.maximum_length	максимальная длина пароля	256	256
securityprofile.minimum_changes	минимальное количество изменений, которое должен содержать новый пароль по сравнению с предыдущим	0	256
securityprofile.failed_login_attempts	количество неудачных попыток входа в СУБД	5	Int_Max
securityprofile.password_lock_time	время, на которое блокируется пользователь в СУБД (1 час в секундах) Значение «-1» означает, что не будет происходить блокировка пользователя на заданное время, т.е. отсутствие блокировки. В противном случае при нарушении политики профиля - пользователь блокируется на указанное в параметре время.	-1	Int_Max
securityprofile.failed_login_attempts_max_time_sec	время в течении которого допустимо ошибиться	0	Int_Max
securityprofile.password_min_life_time	время в секундах, в течение которого пароль должен использоваться и не может быть изменен	0	Int_Max
securityprofile.password_life_time	время в секундах, в течение которого может быть использован текущий пароль (в секундах)	7776000	Int_Max
securityprofile.password_grace_time	время в секундах, в течение которого пользователь может использовать текущий пароль с напоминанием о необходимости его сменить до блокировки аккаунта. Время прибавляется к времени, установленному в	0	Int_Max

Параметр	Примечание	Параметры в профиле	Макс-е значение
	securityprofile.password_life_time		
securityprofile.password_reuse_time	время между повторным использованием одного и того же пароля (в секундах)	0	Int_Max
securityprofile.password_reuse_max	повторное использование пароля	5	Int_Max
securityprofile.store_password_encrypted	хранение паролей в закрытом виде	True	False

6.1.3.8.6 Corporate_3

Профиль парольной политики «Corporate_3» – корпоративный профиль третьего уровня для технических (сервисных, служебных) учетных записей, используемых в технологических процессах ИС или встроенных производителями программных (программно-аппаратных) средств в такие средства.

Таблица 6.7 – Параметры парольной политики «Corporate_3»

Параметр	Примечание	Параметры в профиле	Макс-е значение
Параметры для конфигурирования парольной политики			
securityprofile.profile_name	имя профиля по умолчанию	'Corporate_3'	
securityprofile.special	пароль должен содержать указанные символы	\!"#\$%&()*+ , ./:;<=>?@[]^ _`{ }~	
securityprofile.lower_case_count	пароль должен содержать как минимум 1 символов в нижнем регистре	1	256
securityprofile.upper_case_count	пароль должен содержать как минимум 1 символ в верхнем регистре	1	256
securityprofile.numbers_count	пароль должен содержать как минимум 2 цифры	2	256
securityprofile.special_count	пароль должен содержать как минимум 1 спец. символ из заданного набора special_chars	1	256
securityprofile.minimum_length	минимальная длина пароля	16	256
securityprofile.maximum_length	максимальная длина пароля	256	256

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

Параметр	Примечание	Параметры в профиле	Макс-е значение
securityprofile.minimum_changes	минимальное количество изменений, которое должен содержать новый пароль по сравнению с предыдущим	2	256
securityprofile.failed_login_attempts	количество неудачных попыток входа в СУБД	5	Int_Max
securityprofile.password_lock_time	время, на которое блокируется пользователь в СУБД (1 час в секундах) Значение «-1» означает, что не будет происходить блокировка пользователя на заданное время, т.е. отсутствие блокировки. В противном случае при нарушении политики профиля - пользователь блокируется на указанное в параметре время.	-1	Int_Max
securityprofile.failed_login_attempts_max_time_sec	время в течении которого допустимо ошибиться	0	Int_Max
securityprofile.password_min_life_time	время в секундах, в течение которого пароль должен использоваться и не может быть изменен	30	Int_Max
securityprofile.password_life_time	время в секундах, в течение которого может быть использован текущий пароль (в секундах)	7776000	Int_Max
securityprofile.password_grace_time	время в секундах, в течение которого пользователь может использовать текущий пароль с напоминанием о необходимости его сменить до блокировки аккаунта. Время прибавляется к времени, установленному в securityprofile.password_life_time	-1	Int_Max
securityprofile.password_reuse_time	время между повторным использованием одного и того же пароля (в секундах)	0	Int_Max
securityprofile.password_reuse_max	повторное использование пароля	5	Int_Max

Параметр	Примечание	Параметры в профиле	Макс-е значение
securityprofile.store_password_encrypted	хранение паролей в закрытом виде	True	False

6.1.3.9 Взаимодействие параметров парольных политик

6.1.3.9.1 Время жизни пароля пользователя

Время жизни пароля пользователя состоит из трех параметров:

- максимальное время действия пароля (securityprofile.password_life_time);
- минимальное время действия пароля (securityprofile.password_min_life_time);
- льготный период действия пароля (securityprofile.password_grace_time).

На рисунке 6.1 показана схема взаимодействия временных параметров. На отрезке времени min_life_time пароль пользователя нельзя изменить.

Если временной параметр grace_time активен, то установленное в нем время прибавится к времени параметра life_time и после окончания суммарного срока действия пароля учетная запись пользователя будет заблокирована.

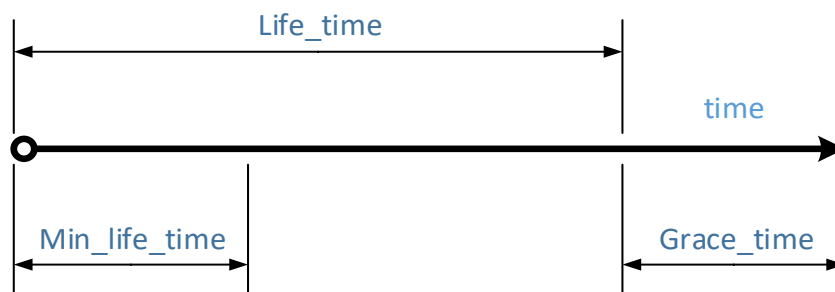


Рисунок 6.1 – Схема взаимодействия временных параметров

6.1.3.9.2 Количество изменений в пароле пользователя

Параметр securityprofile.minimum_changes определяет минимальное количество знаков, которыми должен отличаться новый пароль от предыдущего. Этот параметр взаимосвязан с параметром хранения паролей в закрытом виде securityprofile.store_password_encrypted. По умолчанию параметр securityprofile.store_password_encrypted имеет значение true, т.е. включен.

Если параметр равен false, то будет доступен параметр securityprofile.minimum_changes, при этом аутентификационная информация пользователей будет доступна только роли с атрибутом Superuser.

Повторное использование пароля пользователя осуществляется в несколько этапов, проверяется:

- история паролей; она проверяется вне зависимости от способа хранения аутентификационной информации, указанного в параметре securityprofile.store_password_encrypted;
- заданное время между использованием последнего пароля, указанного в параметре securityprofile.password_reuse_time; если параметр не задан, то проверка пропускается;
- заданный параметр разрешенного количества ранее использованных паролей, указанных в параметре securityprofile.password_reuse_max; если параметр не задан, то проверка пропускается.

Проверки проходят параллельно.

6.1.3.9.3 Повторное использование пароля пользователя

Политика в отношении повторного использования пароля может иметь следующие варианты:

- разрешено установить ранее использованный пароль;
- разрешено установить ранее использованный пароль через определенное время;
- разрешено установить ранее использованный пароль определенное количество раз;
- разрешено установить ранее использованный пароль определенное количество раз через определенное время;
- запрещено установить ранее использованный пароль.

Параметр «securityprofile.password_reuse_time» (промежуток времени, который должен пройти между использованием одинаковых паролей) может принимать три типа значений:

- «-1» – бесконечный промежуток (повторно использовать пароль нельзя);
- «0» – промежуток отсутствует, можно ставить бывший в использовании пароль сразу;
- более «0», но не более максимального значения переменной INT_MAX.

Параметр «securityprofile.password_reuse_max» (минимальное количество смен пароля между использованием одинаковых паролей) может принимать следующие значения:

- «-1» – бесконечное количество (повторно использовать пароль нельзя);
- «0» – количество смен паролей равно нулю, т.е. разрешено использовать предыдущий пароль при текущей смене пароля;
- более «0», но не более максимального значения переменной INT_MAX.

Взаимодействие параметров представлено в матрице.

Таблица 6.8 – Матрица параметров

Повторное использование пароля в течение времени (securityprofile.password_reuse_time)	Количество ранее использованных паролей (securityprofile.password_reuse_max)		
	Запрещено (-1)	Разрешено (0)	Кол-во паролей больше указанного значения
Запрещено (-1)	Запрещено	Запрещено	Запрещено
Разрешено (0)	Запрещено	Разрешено	Разрешено
Период времени (более 0)	Запрещено	Разрешено	Разрешено

Таким образом, для формирования одной из политик следует установить значения для параметров securityprofile.password_reuse_time и securityprofile.password_reuse_max приведенные в таблице 6.9.

Таблица 6.9 – Комбинации параметров значений политик повторного использования пароля

№	Политика	Параметры	Значение
1	Использование ранее использованного пароля запрещено	securityprofile.password_reuse_time	-1
		securityprofile.password_reuse_max	-1
2	Использование ранее использованного пароля запрещено, вне зависимости от количества смен пароля	securityprofile.password_reuse_time	-1
		securityprofile.password_reuse_max	0 (или более 0)
3	Использование ранее использованного пароля разрешено через определенное количество раз смены пароля	securityprofile.password_reuse_time	Более 0
		securityprofile.password_reuse_max	Более 0
4	Использование ранее использованного пароля запрещено. Все пароли уникальные	securityprofile.password_reuse_time	0 (или более 0)
		securityprofile.password_reuse_max	-1
5	Использование ранее использованного пароля разрешено	securityprofile.password_reuse_time	0
		securityprofile.password_reuse_max	0

6.1.3.9.4 Взаимодействие параметров парольных политик при реализации мер защиты информации ИАФ.4 (1г) и УПД.6 (1)

В компоненте SecurityProfile реализована часть требований ИАФ.4 (1г) и УПД.6 (1), описанных в документе «Методический документ. Меры защиты информации в государственных информационных системах» (утв. ФСТЭК России 11.02.2014):

В мере защиты информации ИАФ.4 (1г) пользователь блокируется после достижения установленного максимального количества неудачных попыток аутентификации на период от 15 до 60 минут.

В мере защиты информации УПД.6 (1) пользователь блокируется после достижения установленного максимального количества неудачных попыток аутентификации за установленный период времени до момента разблокирования его администратором.

Для реализации указанных мер используются параметры:

- securityprofile.failed_login_attempts – количество неудачных попыток аутентификации;
- securityprofile.password_lock_time – время блокировки пользователя;
- securityprofile.failed_login_attempts_max_time_sec – время, в течение которого допустимо ошибиться.

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

Ключевым моментом является соотношение временных параметров и количество попыток аутентификации пользователя в разрезе времени.

Далее рассматриваются примеры взаимодействия параметров SecurityProfile.

Пример № 1

Установлены параметры:

- securityprofile.failed_login_attempts = 5 – пять неудачных попыток аутентификации;
- securityprofile.failed_login_attempts_max_time_sec = 0 – время, в течение которого допустимо ошибиться, не установлено;
- securityprofile.password_lock_time = 3600 – время блокирования пользователя 1 час.

Пользователь проводит пять неудачных попыток аутентификации и по параметру securityprofile.password_lock_time = 3600 он блокируется на один час. По истечении времени пользователь может предпринять попытки аутентификации.

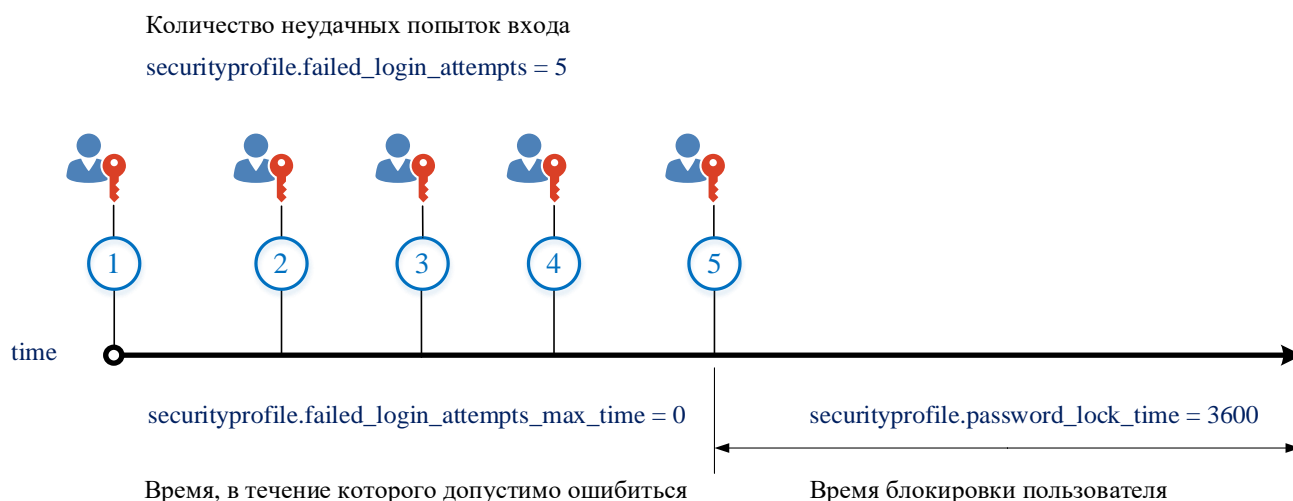


Рисунок 6.2 – Временная диаграмма примера № 1

Таким образом выполняется мера защиты информации ИАФ.4 (1г).

Пример № 2

Установлены параметры:

- securityprofile.failed_login_attempts = 5 – пять неудачных попыток аутентификации;
- securityprofile.failed_login_attempts_max_time_sec = 300 – время, в течение которого допустимо ошибиться, установлено 5 минут;
- securityprofile.password_lock_time = 3600 – время блокирования пользователя 1 час.

Пользователь проводит пять неудачных попыток аутентификации в течение пяти минут. В этом случае параметр password_lock_time игнорируется, и пользователь блокируется до момента разблокирования его администратором.

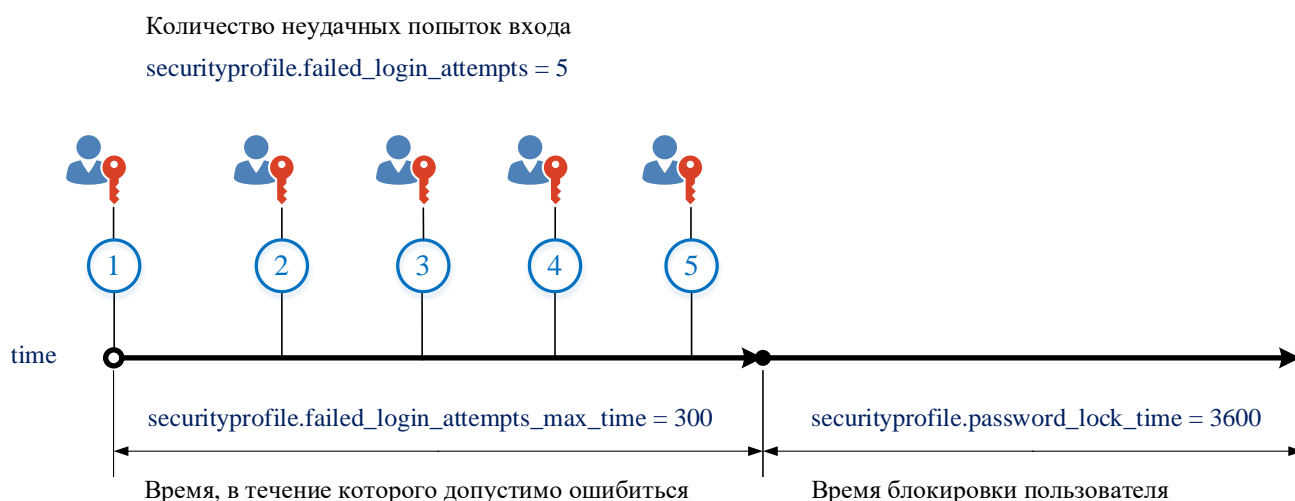


Рисунок 6.3 – Временная диаграмма примера № 2

Таким образом выполняется мера защиты информации УПД.6 (1).

Пример № 3

Установлены параметры:

- securityprofile.failed_login_attempts = 5 – пять неудачных попыток аутентификации;
- securityprofile.failed_login_attempts_max_time_sec = 300 – время, в течение которого допустимо ошибиться, установлено 5 минут;
- securityprofile.password_lock_time = 3600 – время блокирования пользователя 1 час.

Пользователь проводит четыре неудачные попытки аутентификации, но пятая попытка аутентификации удачная и проходит после времени, в течение которого допустимо ошибиться. Пользователь не блокируется, т.к. не превышены установленные параметры.

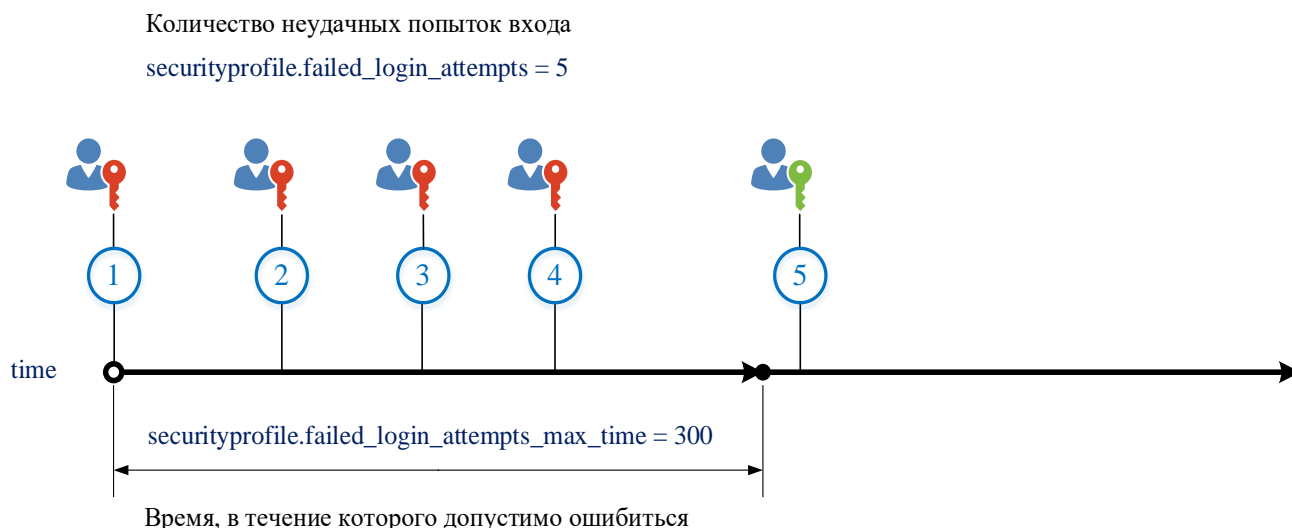


Рисунок 6.4 – Временная диаграмма примера № 3

Пример № 4

Установлены параметры:

- `securityprofile.failed_login_attempts = 5` – пять неудачных попыток аутентификации;
- `securityprofile.failed_login_attempts_max_time_sec = 300` – время, в течение которого допустимо ошибиться, установлено 5 минут;
- `securityprofile.password_lock_time = 3600` – время блокирования пользователя 1 час.

Пользователь проводит четыре неудачные попытки аутентификации. Пятая неудачная попытка происходит после истечения времени, установленного в параметре `securityprofile.failed_login_attempts_max_time_sec`, т.е. с момента первой неудачной аутентификации прошло более пяти минут.

Далее срабатывает параметр `securityprofile.password_lock_time` и пользователь блокируется на один час. Вмешательство администратора не потребуется, и пользователь сможет предпринять попытки аутентификации через час с момента пятой неудачной попытки аутентификации.

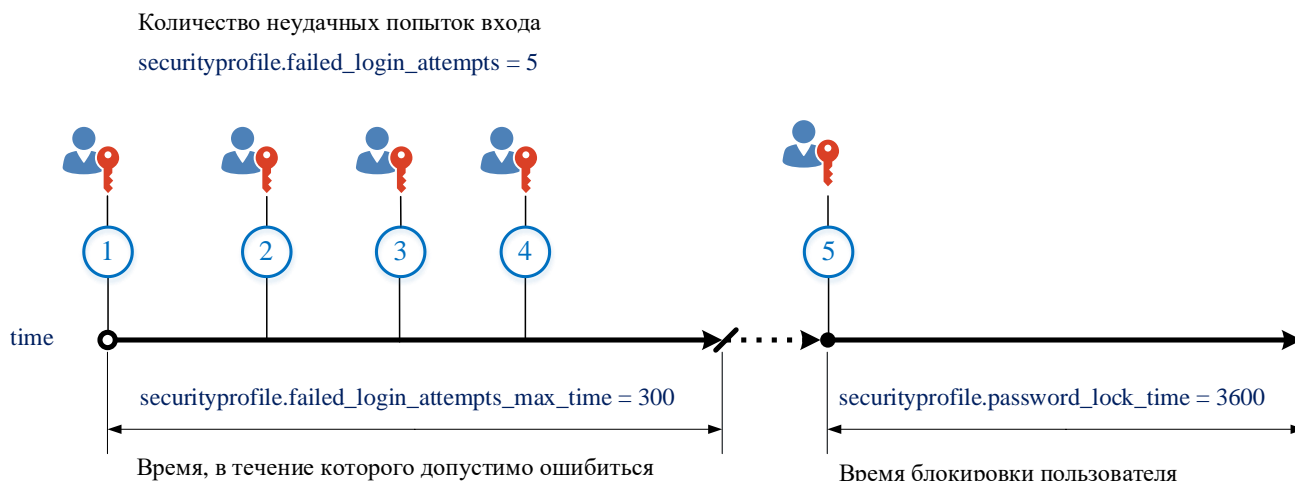


Рисунок 6.5 – Временная диаграмма примера № 4

Таким образом выполняется мера защиты информации ИАФ.4 (1г).

6.1.3.10 Параметры парольной политики по схеме securityprofile

Для активации парольной политики по схеме securityprofile в СУБД «Jatoba» необходимо выполнить следующие действия:

- 1) От учетной записи администратора СУБД подключиться к ОС и открыть файл postgresql.conf³⁾.
- 2) В файле postgresql.conf прописать следующую строку:

```
shared_preload_libraries = 'securityprofile'
```

- 3) Перезагрузить сервис СУБД «Jatoba» в:

- ОС Windows Server при помощи команд:

```
net stop JatobaServer;
net start JatobaServer
```

- ОС семейства GNU/Linux при помощи команды:

```
systemctl restart jatoba-<ver>
```

³⁾ Местонахождение файла postgresql.conf по умолчанию в:

- ОС семейства GNU/Linux: /var/lib/jatoba/<ver>/data/postgresql.conf;
- ОС Windows Server: C:\Program Files\GIS\Jatoba\<ver>\data\postgresql.conf.



Необходимо обратить внимание, что указание в конфигурационном файле postgresql.conf опции shared_preload_libraries = 'securityprofile' активирует компонент управления парольными политиками и создает политику по умолчанию с именем «default». Для создания новых политик выполняется п. г)

4) От учетной записи администратора СУБД подключиться к БД, указанной в конфигурационном файле postgresql.conf в параметре securityprofile.db_name, и затем выполнить следующую команду:

```
CREATE EXTENSION securityprofile;
```

После чего сформируется служебная схема securityprofile (см. рисунок 6.6) для указанной БД.



Для корректной работы компонента SecurityProfile допустима единственная установка расширения SecurityProfile для СУБД.

```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
postgres=# create extension securityprofile;
CREATE EXTENSION
postgres=#
```

Рисунок 6.6 – Установка расширения «securityprofile»

Будет создан профиль по умолчанию с именем «default».

В таблице 6.10 представлены команды управления параметрами служебной схемой в СУБД «Jatoba».

Таблица 6.10 – Команды управления параметрами профиля в СУБД «Jatoba»

Команда	Примечание
Количество неуспешных попыток аутентификации	
select securityprofile.set_profile_failed_login_attempts('имя_профиля', bigint);	Установка максимального количества неудачных попыток входа в СУБД. Если порог превышен, то применяется временная блокировка
select securityprofile.set_profile_failed_login_attempts_max_time_seconds('имя_профиля', bigint);	Установка времени, в течение которого пользователь навсегда блокируется при достижении установленного максимального количества неуспешных попыток аутентификации
№ изменения: _____	Подпись отв. лица: _____ Дата внесения изм: _____

Команда	Примечание
select securityprofile.set_profile_password_lock_time (имя_профиля', bigint);	Установка времени, на которое блокируется пользователь в днях, -1 – вечная блокировка
select securityprofile.set_profile_password_lock_time_s econds(имя_профиля', bigint);	Установка времени, на которое блокируется пользователь в секундах
Время действия пароля	
select securityprofile.set_profile_password_min_life_ti me(имя_профиля', bigint);	Установка времени в днях, в течение которого пароль должен использоваться и не может быть изменен
select securityprofile.set_profile_password_min_life_ti me_seconds(имя_профиля', bigint);	Установка минимального времени в секундах, в течение которого пароль должен использоваться и не может быть изменен
select securityprofile.set_profile_password_life_time('и мя_профиля', bigint);	Установка времени в днях, в течение которого может быть использован текущий пароль (в днях). (-1 – вечность. Значение, указанное в опции VALID UNTIL при создании/смене пароля будет проигнорировано)
select securityprofile.set_profile_password_life_time_se conds(имя_профиля', bigint);	Установка времени в секундах, в течение которого может быть использован текущий пароль
select securityprofile.set_profile_password_grace_time (имя_профиля', bigint);	Установка времени в днях, в течение которого пользователь может использовать текущий пароль с напоминанием о необходимости его сменить до блокировки аккаунта. (-1 – вечность. Аккаунт суперпользователя заблокирован не будет)
select securityprofile.set_profile_password_grace_time_ seconds(имя_профиля', bigint);	Установка времени в секундах, в течение которого пользователь может использовать текущий пароль с напоминанием о необходимости его сменить до блокировки аккаунта. (-1 – вечность. Аккаунт суперпользователя заблокирован не будет)
Повторное использование пароля	
select securityprofile.set_profile_password_reuse_time (имя_профиля', bigint);	Установка времени в днях между повторным использованием одного и того же пароля. (-1 – вечность)
select securityprofile.set_profile_password_reuse_time_ seconds(имя_профиля', bigint);	Установка времени в секундах между повторным использованием одного и того же пароля.
select securityprofile.set_profile_password_reuse_max('и мя_профиля', integer);	Установка количества смен пароля перед возвращением к старому значению (Для повторного использования пароля без ограничений надо указать password_reuse_time=-1 и password_reuse_max = -1)
Характеристики пароля	

Команда	Примечание
select securityprofile.set_profile_password_lower_case_count('имя_профиля', integer);	Установка минимального количества символов в нижнем регистре, которые должен содержать пароль. (0 – наличие не обязательно)
select securityprofile.set_profile_password_upper_case_count('имя_профиля', integer);	Установка минимального количества символов в верхнем регистре, которые должен содержать пароль. (0 – наличие не обязательно)
select securityprofile.set_profile_password_numbers_count('имя_профиля', integer);	Установка параметра минимального количества цифр, которые должен содержать пароль (0 – наличие не обязательно)
select securityprofile.set_profile_password_special_chars ('имя_профиля', 'набор спец.символов');	Параметр указывающий на перечень набора спец.символов, которые должны использоваться в пароле
select securityprofile.set_profile_password_special_count('имя_профиля', integer);	Установка параметра, при котором пароль должен содержать минимальное количество спецсимволов (0 – наличие не обязательно)
select securityprofile.set_profile_password_min_len('имя_профиля', integer);	Установка минимальной длины пароля
select securityprofile.set_profile_password_max_len('имя_профиля', integer);	Параметр устанавливающий максимальную длину пароля
select securityprofile.set_profile_password_min_change_s('имя_профиля', integer);	Параметр устанавливающий минимальное количество изменений, которое должен содержать новый пароль по сравнению с предыдущим. (Работает только при условии хранения истории паролей в открытом виде)
select securityprofile.set_profile_store_password_encrypted('имя_профиля', boolean) – true	Параметр, устанавливающий хранение истории паролей в закрытом виде (в md5 хэшах), false – в открытом

Кроме того, есть функциональная возможность создавать дополнительные схемы и привязывать к ним учетные записи пользователей.

В таблице 6.11 представлены команды управления профилем в СУБД «Jatoba».

Таблица 6.11 – Команды управления профилем в СУБД «Jatoba»

Команда	Примечание
select securityprofile.create_profile ('имя_нового_профиля');	Создание нового профиля
select * from securityprofile.show_profiles;	Просмотр списка профилей
select securityprofile.drop_profile ('имя_профиля');	Удаление профиля
select securityprofile.bind_profile ('имя_профиля', 'имя_пользователя');	Привязка пользователя к профилю



Необходимо обратить внимание, что для применения схемы «securityprofile» к ранее созданным учетным записям, необходимо выполнить команду:

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

```
bind_profile('имя_профиля', 'имя_пользователя')
```

Пример1. Создание профиля парольной политики

Задание. Требуется создать дополнительный профиль парольной политики с именем «password_user», в котором будет установлена длина пароля пользователя равная 9 символам. При этом пользователь «test1» должен быть привязан к профилю «password_user».

Шаг 1. Создать профиль парольной политики с именем «password_user» на основании схемы «securityprofile», выполнив следующую команду:

```
SELECT securityprofile.create_profile('password_user');
```

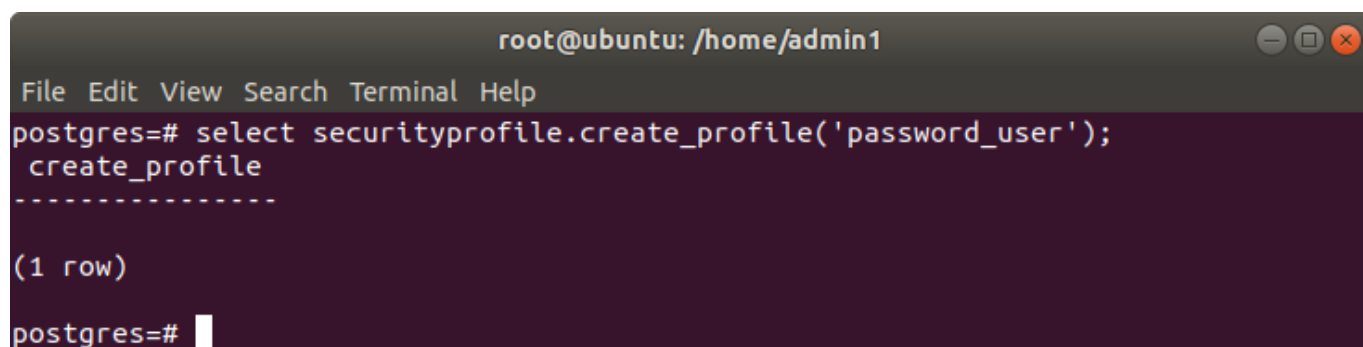


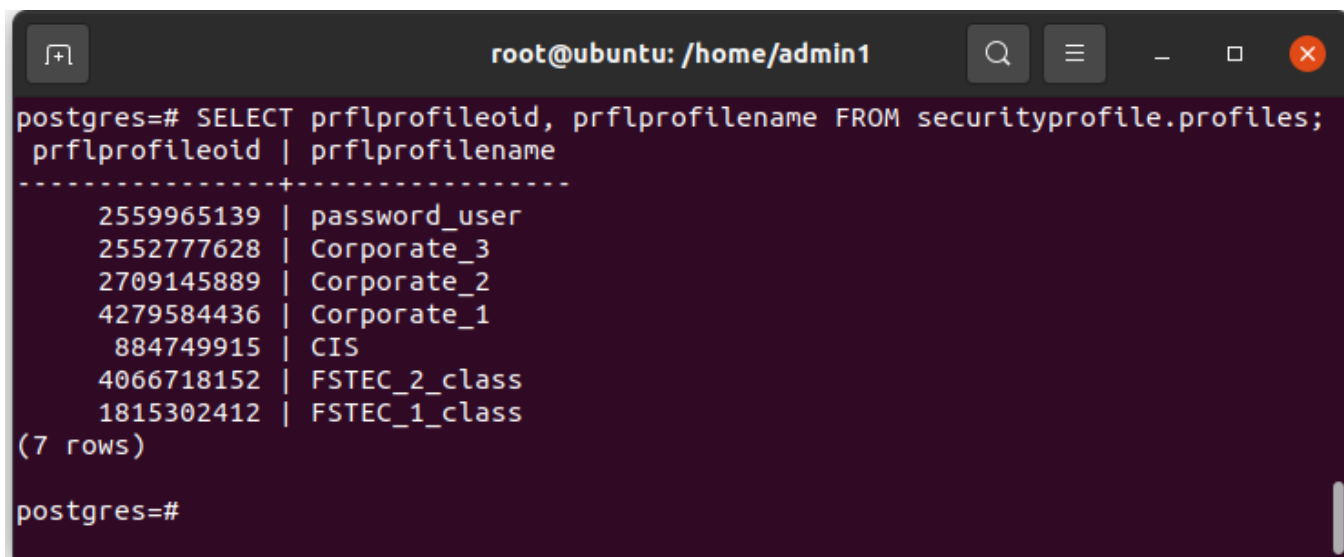
Рисунок 6.7 – Создание профиля «password_user»

Шаг 2. Просмотреть список имеющихся профилей, выполнив SQL-команду:

```
SELECT * from securityprofile.show_profiles;
```

либо просмотреть несколько столбцов из таблицы «securityprofile.profiles», выполнив SQL-команду:

```
SELECT prflprofileoid, prflprofilename FROM  
securityprofile.profiles;
```



```

root@ubuntu: /home/admin1
postgres=# SELECT prflprofileoid, prflprofilename FROM securityprofile.profiles;
 prflprofileoid | prflprofilename
-----+-----
      2559965139 | password_user
      2552777628 | Corporate_3
      2709145889 | Corporate_2
      4279584436 | Corporate_1
       884749915 | CIS
      4066718152 | FSTEC_2_class
      1815302412 | FSTEC_1_class
(7 rows)

postgres=#

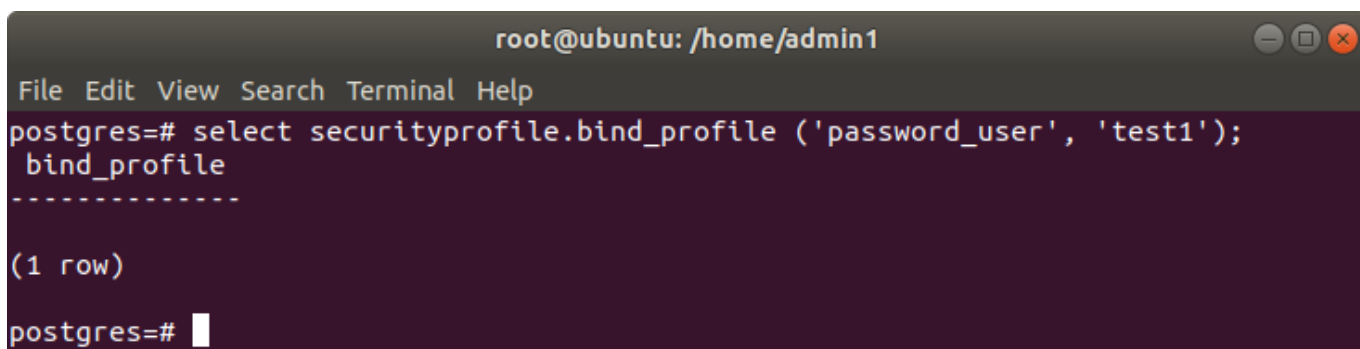
```

Рисунок 6.8 – Список профилей парольных политик

В результате видно, что новый профиль создан и унаследовал параметры парольной политики по умолчанию.

Шаг 3. Привязать пользователя «test1» к профилю «password_user», выполнив команду:

```
SELECT securityprofile.bind_profile ('password_user', 'test1');
```



```

root@ubuntu: /home/admin1
File Edit View Search Terminal Help
postgres=# select securityprofile.bind_profile ('password_user', 'test1');
 bind_profile
-----
(1 row)

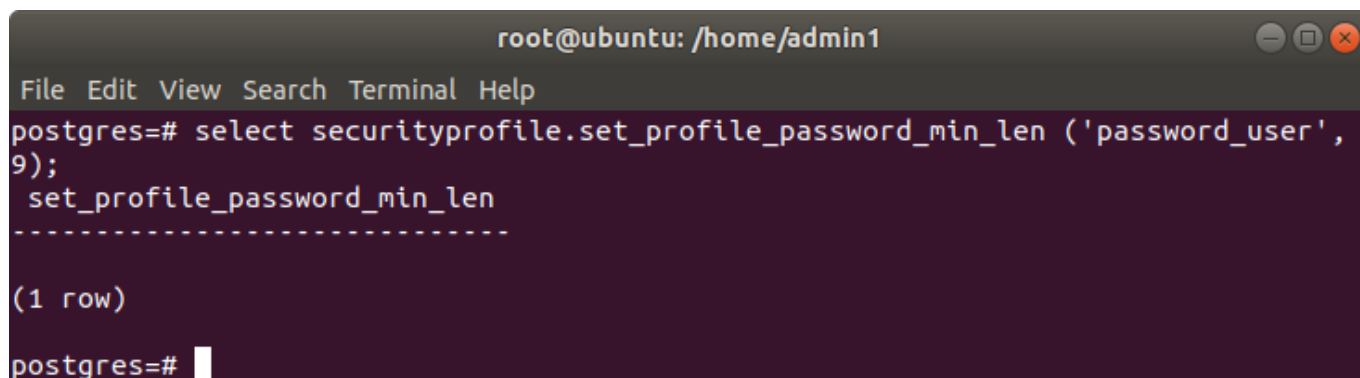
postgres=#

```

Рисунок 6.9 – Привязка пользователя к профилю

Шаг 4. Установить минимальную длину пароля пользователя равную 9 символам, выполнив команду:

```
SELECT securityprofile.set_profile_password_min_len
('password_user', 9);
```



```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
postgres=# select securityprofile.set_profile_password_min_len ('password_user',
9);
 set_profile_password_min_len
-----
(1 row)
postgres=#
```

Рисунок 6.10 – Установка длины пароля в профиле «password_user»

На данном шаге требуемое задание выполнено.

6.1.4. Удаление компонента «securityprofile»

Для удаления модуля компонента «securityprofile» из СУБД «Jatoba» необходимо выполнить следующие действия:

1) От учетной записи администратора СУБД подключиться к ОС и открыть файл postgresql.conf.

! **Важно!** В конфигурационном файле postgresql.conf из параметра shared_preload_libraries необходимо удалить значение securityprofile.

2) Авторизоваться в СУБД с учетной записью привилегированного пользователя и выполнить команду:

```
DROP EXTENSION securityprofile;
```

В процессе удаления расширения также выполняется удаление схемы «securityprofile» из БД.

3) Выполнить перезагрузку службы СУБД при помощи команд:

```
systemctl stop jatoba-<ver>
systemctl start jatoba-<ver>
```

4) Проверить состояние службы СУБД:

```
systemctl status jatoba-<ver>
```

6.1.5. Взаимодействие с компонентом управления кластером «ja_Dog»

Компонент «securityprofile» может использоваться с компонентом управления кластером «ja_Dog».

При таком использовании на резервном узле после выполнения репликации, в конфигурационный файл «postgresql.conf» необходимо добавить параметр:

```
securityprofile.sync_delay
```

со значением от «0» до «Int_Max».

Параметр задерживает синхронизацию кеша расширения.

Значение «0» соответствует отключению синхронизации кеша расширения.

Значения от «0» до «Int_Max» задают время в миллисекундах синхронизации кеша расширения. Такие значения должны быть целыми, положительными числами.

После установки параметра следует перезагрузить кластер.

При последующих ручных или автоматических сменах ролей узлов следует вручную на новом главном узле в конфигурационном файле закомментировать или удалить строку с добавленным параметром «securityprofile.sync_delay», а на резервном – добавить.

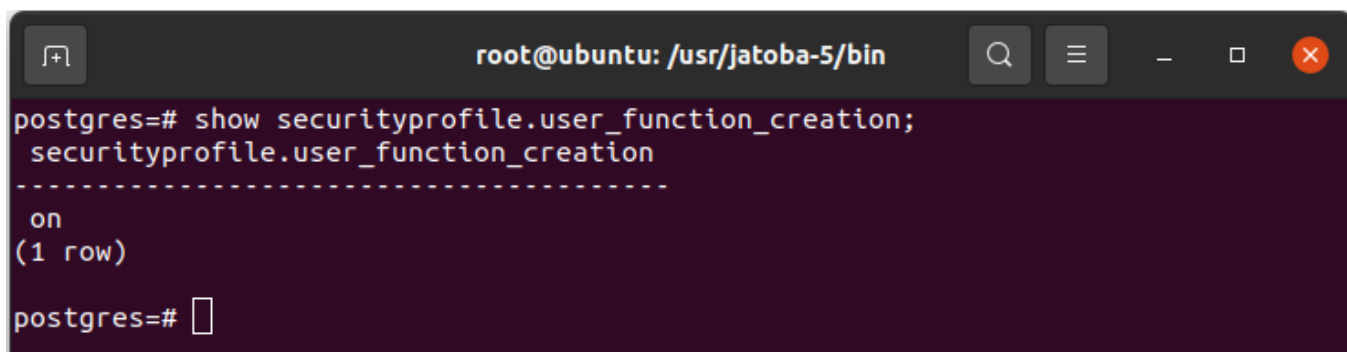
6.1.6. Взаимодействие с компонентом контроля целостности «ja_CSum»

Компонент «securityprofile» может использоваться с компонентом контроля целостности «ja_CSum» для вызова функций блокирования групп пользователей (см. п. 6.2.1). Функции блокировки вызываются автоматически, когда компонентом «ja_CSum» обнаружен в СУБД компонент «securityprofile».

Помимо этого, компонент «securityprofile» обладает функциональной возможностью блокировки установки пользовательских функций.

Для активации этой функциональной возможности требуется убедиться, что функция блокировки установки пользовательских функций у компонента «securityprofile» включена. При выводе должно быть значение «ON».

```
SHOW securityprofile.user_function_creation;
```



```
root@ubuntu: /usr/jatoba-5/bin
postgres=# show securityprofile.user_function_creation;
securityprofile.user_function_creation
-----
on
(1 row)
postgres=#
```

Рисунок 6.11 – Вывод статуса функции блокировки создания функций пользователями

6.1.7. Взаимодействие с компонентом JDV

Компонент «securityprofile» может использоваться с компонентом контроля субъектов доступа «Jatoba data vault» (JDV).

При использовании совместно с компонентом JDV устанавливается время истечения срока действия пароля для следующих служебных учетных записей JDV:

- роль «dv_owner» - администратор безопасности защищаемых таблиц и пользователей (Security administrator);
- роль «dv_acctmgr» - администратор пользователей (User administrator).

Для корректной работы данных ролей необходимо создать отдельный профиль парольной политики в соответствии с следующими шагами:

- 1) Создать специальный служебный профиль парольной политики:

```
SELECT securityprofile.create_profile(' dv_profile');
```

- 2) Сделать бессрочный срок действия пароля для профиля:

```
SELECT
securityprofile.set_profile_password_life_time('dv_profile', -
1);
```


- 3) Сделать бессрочный льготный срок действия пароля для профиля:

```
SELECT  
securityprofile.set_profile_password_grace_time('dv_profile', -  
1);
```

- 4) Установить бессрочное время простоя (отсутствия активности) профиля:

```
SELECT  
securityprofile.set_profile_user_idle_days_max('dv_profile', -  
1);
```

- 5) После установки расширения JDV выполнить добавление ролей dv_acctmgr и dv_owner к профилю dv_profile:

```
SELECT securityprofile.bind_profile('dv_profile', 'dv_owner');  
SELECT securityprofile.bind_profile('dv_profile',  
'dv_acctmgr');
```

6.1.8. Взаимодействие с компонентом ja_hipe_cluster/Citus

Компонент SecurityProfile может использоваться с компонентом контроля субъектов доступа «ja_hipe_cluster/Citus».

Расширение «citus» необходимо устанавливать в БД, в которой уже установлено расширение «securityprofile».

При совместном применении SecurityProfile с компонентом ja_hipe_cluster/Citus необходимо перевести обработку изменения пароля пользователя, получаемого от узла-координатора на рабочий узел в хешированном виде, при помощи внесения изменений в конфигурационный файл «postgresql.conf»:

```
securityprofile.hash_password_strictness = off
```

После установки параметра следует перезагрузить СУБД на всех узлах кластера, на которых изменялся параметр:

```
systemctl restart jatoba-6
```

В случае, если параметр `securityprofile.hash_password_strictness` не изменен в соответствии с указанными выше рекомендациями, при добавлении рабочего узла (worker) в распределенный кластер компонента `ja_hipe_cluster/Citus`, будет возникать следующая ошибка:

ОШИБКА: Can not check password validity.

ПОДРОБНОСТИ: Password is hashed.

```
ОПЕРАТОР: SELECT worker_create_or_atelier_role('postgres' 'CREATE
ROLE postgres SUPERUSER CREATEDB CREATEROLE INHERIT LOGIN
REPLICATION BYPASSRLS CONNECTION LIMIT -1 PASSWORD'...)
```

6.2. Управление доступом субъектов доступа к объектам доступа

6.2.1. Блокирование и разблокирование учетных записей

Блокирование учетных записей СУБД выполняется функциональными возможностями компонента «securityprofile».

Блокирование/разблокирование УЗ может выполняться для:

- отдельного пользователя (см. п. 6.2.1.1);
- группы пользователей (см. п. 6.2.1.2);
- группы пользователей Администраторы БД (см. п. 6.2.1.3).

По умолчанию блокировка пользователей выполняется в режиме «immediate». В данном режиме пользователь принудительно отключается без ожидания и непосредственного отката транзакций.

6.2.1.1 Блокирование/разблокирование учетной записи пользователя

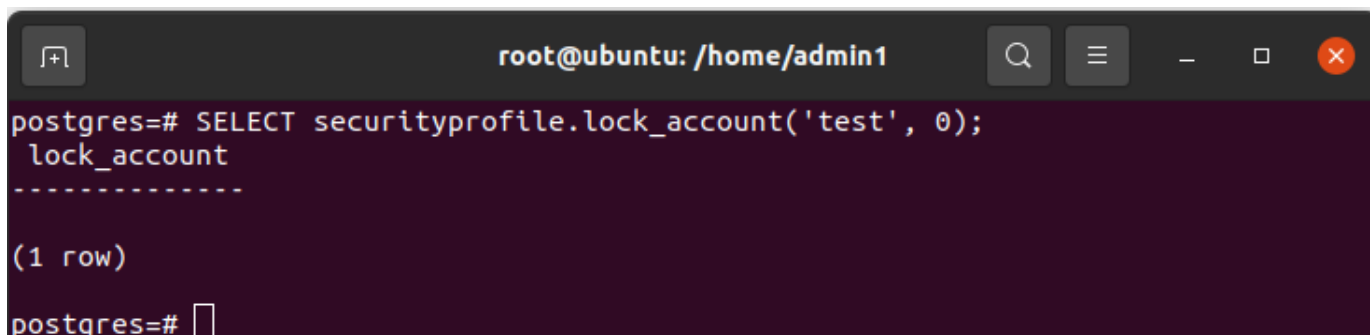
Для блокировки учетной записи пользователя необходимо администратору СУБД выполнить SQL-команду с синтаксисом:

```
SELECT securityprofile.lock_account('имя_пользователя',
bigint);
```

Например

Заблокируем учетную запись пользователя СУБД «test» SQL-командой:

```
SELECT securityprofile.lock_account('test', 0);
```



The screenshot shows a terminal window with the title bar 'root@ubuntu: /home/admin1'. The command 'postgres=# SELECT securityprofile.lock_account('test', 0);' has been entered. The output shows 'lock_account' followed by a dashed line and '(1 row)'. The prompt 'postgres=#' is visible at the bottom.

Рисунок 6.12 – Блокировка пользователя

Примечание: bigint – задержка, с которой будет выполнена блокировка в днях.

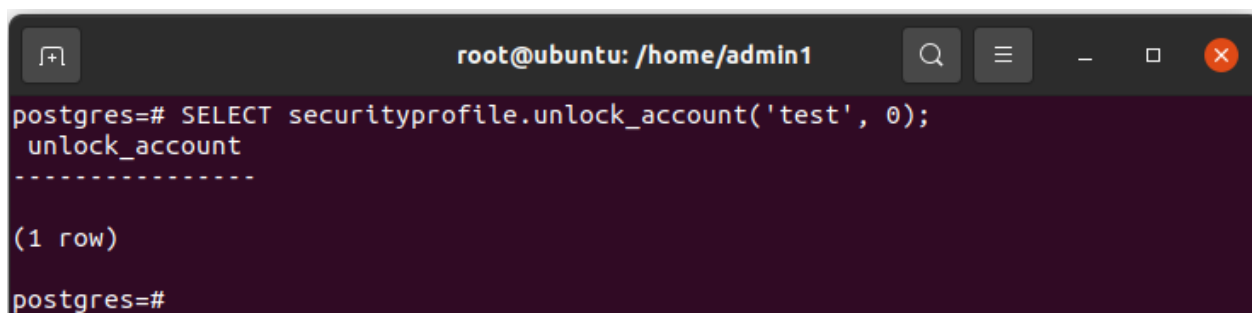
select lock_account_seconds('имя_пользователя', bigint) – задержка, с которой будет выполнена блокировка в секундах.

Для разблокировки учетных записей пользователей администратору СУБД необходимо выполнить следующую команду:

```
SELECT securityprofile.unlock_account ('имя_пользователя',  
bigint);
```

Примечание: bigint – задержка, с которой будет выполнено снятие блокировки в днях.

```
SELECT securityprofile.unlock_account('test', 0);
```



The screenshot shows a terminal window with the title bar 'root@ubuntu: /home/admin1'. The command 'postgres=# SELECT securityprofile.unlock_account('test', 0);' has been entered. The output shows 'unlock_account' followed by a dashed line and '(1 row)'. The prompt 'postgres=#' is visible at the bottom.

Рисунок 6.13 – SQL-команда блокирования пользователя

unlock_account_seconds ('имя_пользователя', bigint) – задержка, с которой будет выполнено снятие блокировки в секундах.

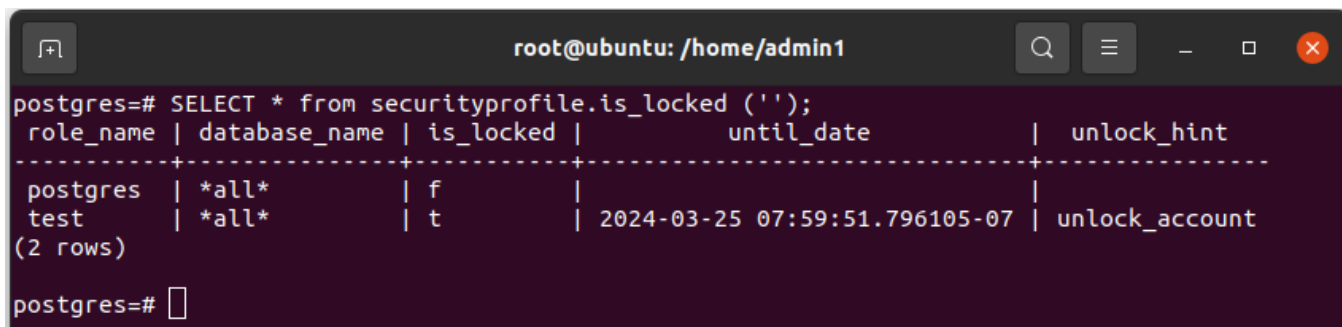
Для проверки факта блокировки и времени, в течение которого она будет действовать, администратору СУБД необходимо выполнить следующую команду:

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

```
SELECT * from securityprofile.is_locked('имя_пользователя');
```

Вывод информации о всех пользователях выполняется SQL-командой:

```
SELECT * from securityprofile.is_locked ('');
```



```
root@ubuntu: /home/admin1
postgres=# SELECT * from securityprofile.is_locked ('');
role_name | database_name | is_locked |          until_date          | unlock_hint
-----+-----+-----+-----+-----
postgres  | *all*         | f         |                               | 
test      | *all*         | t         | 2024-03-25 07:59:51.796105-07 | unlock_account
(2 rows)
postgres=#
```

Рисунок 6.14 – Вывод списка состояния пользователей

В выводе команды присутствуют столбцы:

- role_name – список заблокированных пользователей;
- database_name – база данных;

Отображаются значения «all», если пользователь заблокирован на основании нарушений парольных политик, которые запрещают вход во все базы данных СУБД или имя базы данных, вход в которую пользователю заблокирован.

- is_locked – статус блокировки пользователя;

Отображаются значения «t» (true) – заблокирован и «f» (false) – не заблокирован.

- until_date – дата и время блокировки;

У незаблокированного пользователя поле будет пустым. Также в этом столбце может выводиться значение «infinity», обозначающее вечную блокировку.

- unlock_hint – подсказка по разблокировке.

Отображает подсказку администратору СУБД о том, какую функцию securityprofile использовать для разблокировки пользователя.

Поле содержит значение «resume_users», если блокировка выполнена из-за нарушения контроля целостности СУБД по директиве компонента «ja_CSum».

Значение «unlock_account» отображается, если пользователь заблокирован принудительно администратором СУБД или при нарушении парольных политик

6.2.1.2 Блокирование/разблокирование группы учетных записей пользователей

Блокировка группы учетных записей пользователей выполняется тремя функциями:

- suspend_users – блокировка группы пользователей;
- suspend_users_seconds – блокировка группы пользователей с задержкой в секундах;
- suspend_users_noerror – блокировка группы пользователей с игнорированием ошибки ранее заблокированных пользователей.

6.2.1.2.1 Блокировка группы пользователей (securityprofile.suspend_users)

Функция «suspend_users» позволяет заблокировать категорию пользователей для заданной базы данных и имеет синтаксис SQL-команды:

```
suspend_users(bigint, text DEFAULT NULL);
```

Если второй параметр = NULL, то время отсрочки блокировки задано в днях.

Например

Блокировка пользователей БД «db_test» выполняется SQL-командой:

```
SELECT securityprofile.suspend_users('db_test', 0);
```

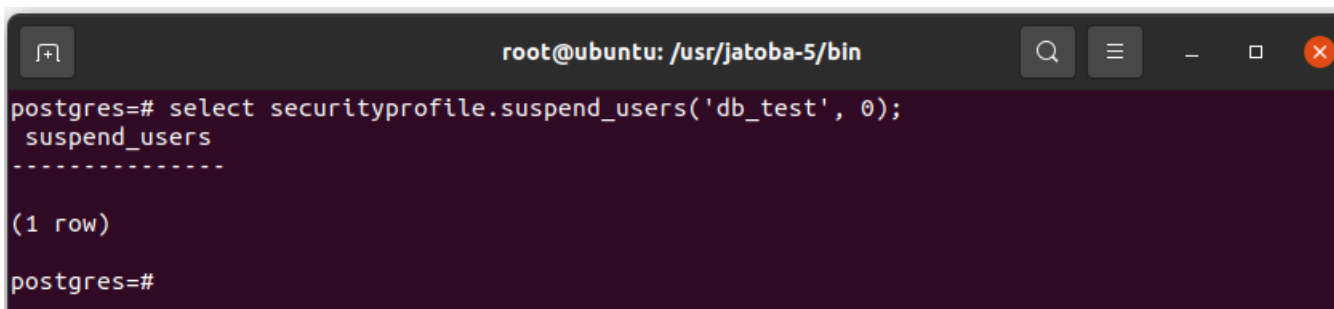


Рисунок 6.15 – Блокировка пользователей БД «db_test»

При проверке статуса блокировки, будет добавлена строка с OID БД «16909».

```
SELECT * FROM securityprofile.status;
```

```

root@ubuntu: /usr/jatoba-5/bin
postgres=# SELECT * FROM securityprofile.status;
 databaseoid |      suspendeduserstime      | suspendedadminstime
-----+-----+-----
          5 | 2023-11-10 03:29:46.377456-08 | 1999-12-31 16:00:00-08
          1 | 2023-11-10 03:29:46.37765-08  | 1999-12-31 16:00:00-08
          4 | 2023-11-10 03:29:46.377705-08 | 1999-12-31 16:00:00-08
       16909 | 2023-11-13 06:39:34.380913-08 | 1999-12-31 16:00:00-08
(4 rows)

postgres=#

```

Рисунок 6.16 – Проверка статуса блокировки

Данный OID БД «16909» принадлежит БД «db_test», что проверяется SQL-командой:

```
SELECT oid FROM pg_database WHERE datname = 'db_test';
```

```

root@ubuntu: /usr/jatoba-5/bin
postgres=# SELECT oid FROM pg_database WHERE datname = 'db_test';
 oid
-----
 16909
(1 row)

postgres=#

```

Рисунок 6.17 – Команда проверки OID БД

6.2.1.2.2 Разблокирование группы пользователей (securityprofile.resume_users)

Разблокировка пользователей выполняется функцией «resume_users» имеющей синтаксис SQL-команды:

```
resume_users(bigint, text DEFAULT NULL);
```

Например

Разблокируем пользователей БД «db_test» SQL-командой:

```
SELECT securityprofile.resume_users('db_test', 0);
```

```

root@ubuntu: /usr/jatoba-5/bin
postgres=# select securityprofile.resume_users('db_test', 0);
 resume_users
-----
(1 row)
postgres=#

```

Рисунок 6.18 – Команда разблокировки пользователей БД

Проверка статуса блокировки покажет изменение даты в поле «suspendeduserstime».

```

root@ubuntu: /usr/jatoba-5/bin
postgres=# select securityprofile.resume_users('db_test', 0);
 resume_users
-----
(1 row)

postgres=# SELECT * FROM securityprofile.status;
 databaseoid | suspendeduserstime | suspendedadminstime
-----
          5 | 2023-11-10 03:29:46.377456-08 | 1999-12-31 16:00:00-08
          1 | 2023-11-10 03:29:46.37765-08 | 1999-12-31 16:00:00-08
          4 | 2023-11-10 03:29:46.377705-08 | 1999-12-31 16:00:00-08
       16909 | 1976-02-17 09:56:07.269431-08 | 1999-12-31 16:00:00-08
(4 rows)

postgres=#

```

Рисунок 6.19 – Статус блокировки пользователей

6.2.1.2.3 Блокировка группы пользователей с задержкой блокировки в устанавливаемой в секундах (securityprofile.suspend_users_seconds)

Функция «suspend_users_seconds» служит для блокировки пользователей БД с задержкой блокировки и имеет синтаксис SQL-команды:

```
suspend_users_seconds(bigint, text DEFAULT NULL);
```

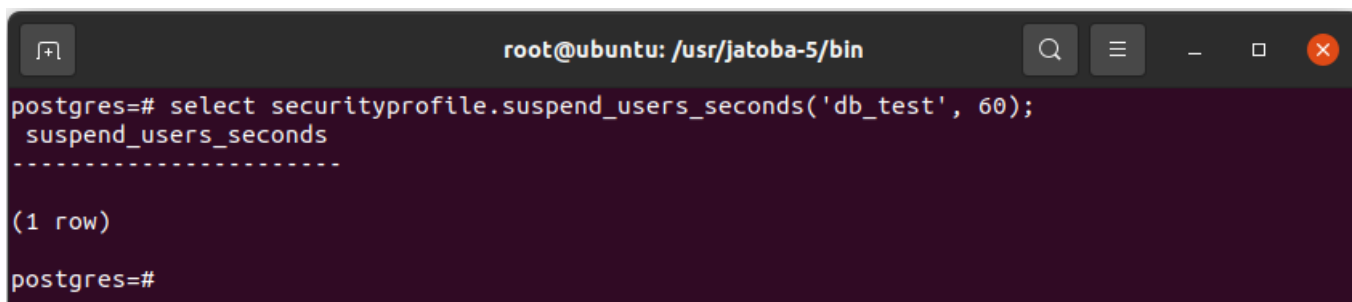
Например

Блокирование пользователей БД «db_test» с отсрочкой блокировки устанавливаемой в секундах, выполняется SQL-командой:

```
SELECT securityprofile.suspend_users_seconds('db_test', 60);
```

В SQL-команде указывается БД и время задержки блокировки, устанавливаемой в секундах.

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



```
root@ubuntu: /usr/jatoba-5/bin
postgres=# select securityprofile.suspend_users_seconds('db_test', 60);
suspend_users_seconds
-----
(1 row)
postgres=#
```

Рисунок 6.20 – Блокирование пользователей БД с отсрочкой блокировки устанавливаемой в секундах

6.2.1.2.4 Разблокирование группы пользователей с задержкой блокировки в устанавливаемой в секундах (securityprofile.resume_users_seconds)

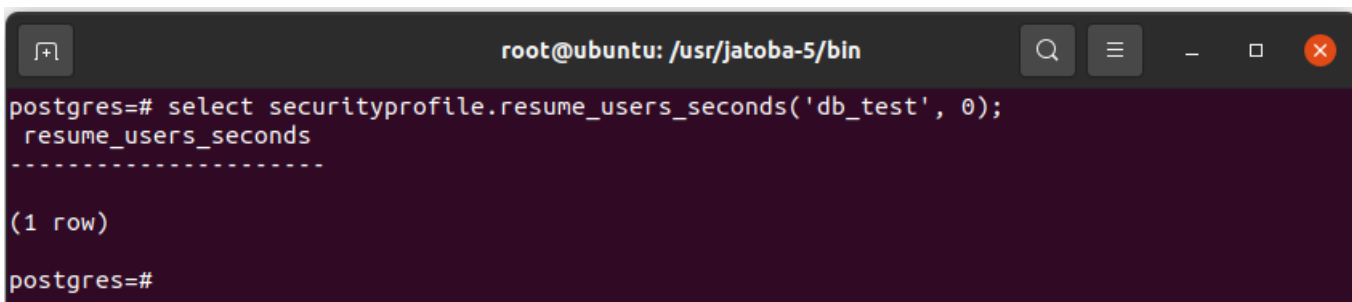
Разблокирование пользователей выполняется функцией «resume_users_seconds» имеющей синтаксис:

```
resume_users_seconds(bigint, text DEFAULT NULL);
```

Например

Разблокируем пользователей БД «db_test» с задержкой в секундах, SQL-командой:

```
SELECT securityprofile.resume_users_seconds('db_test', 0);
```



```
root@ubuntu: /usr/jatoba-5/bin
postgres=# select securityprofile.resume_users_seconds('db_test', 0);
resume_users_seconds
-----
(1 row)
postgres=#
```

Рисунок 6.21 – Команда разблокировки пользователей БД с установленной задержкой по времени

Проверка статуса блокировки покажет изменение даты в поле «suspendeduserstime».

```
SELECT * FROM securityprofile.status;
```



```

root@ubuntu: /usr/jatoba-5/bin

postgres=# select securityprofile.resume_users_seconds('db_test', 0);
resume_users_seconds
-----
(1 row)

postgres=# SELECT * FROM securityprofile.status;
 databaseoid | suspendeduserstime | suspendedadminstime
-----+-----+-----
          5 | 2023-11-10 03:29:46.377456-08 | 1999-12-31 16:00:00-08
          1 | 2023-11-10 03:29:46.37765-08 | 1999-12-31 16:00:00-08
          4 | 2023-11-10 03:29:46.377705-08 | 1999-12-31 16:00:00-08
       16909 | 1976-02-17 08:28:11.33161-08 | 1999-12-31 16:00:00-08
(4 rows)

postgres=#

```

Рисунок 6.22 – Просмотр состояния блокировки

6.2.1.2.5 Блокировка группы пользователей с игнорированием ошибки (securityprofile.suspend_users_noerror)

Функция «suspend_users_noerror» аналогично suspend_users, только не выдает ошибки при наличии уже установленной блокировки и имеет синтаксис SQL-команды:

```
suspend_users_noerror(text, bigint);
```

Например

Блокировка пользователей БД «db_test» игнорированием ошибки, выполняется SQL-командой:

```
SELECT securityprofile.suspend_users_noerror ('db_test', 0);
```

```

root@ubuntu: /usr/jatoba-5/bin

postgres=# select securityprofile.suspend_users_noerror ('db_test', 0);
WARNING: securityprofile: Users were suspended from database db_test already.
suspend_users_noerror
-----
(1 row)

postgres=#

```

Рисунок 6.23 – Блокировка пользователей БД «db_test» игнорированием ошибки

В результате выполнения SQL-команды компонент заблокирует пользователей указанной БД и выведет сообщение, в котором сообщается, что «Пользователи уже были отключены от базы данных»:

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

Users were suspended from database db_test already.

6.2.1.2.6 Разблокировка группы пользователей с игнорированием ошибки (resume_users_noerror)

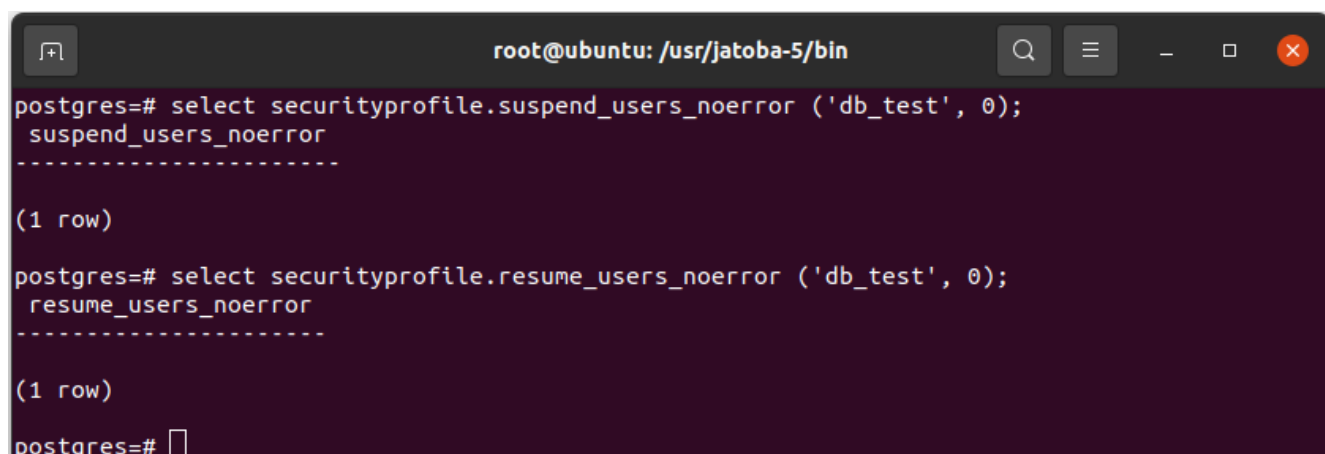
Разблокирование пользователей выполняется функцией «resume_users_noerror» имеющей синтаксис:

```
resume_users_noerror(text, bigint);
```

Например

Разблокируем пользователей БД «db_test», SQL-командой вне зависимости от имеющихся ошибок:

```
SELECT securityprofile.resume_users_noerror ('db_test', 0);
```



```
root@ubuntu: /usr/jatoba-5/bin
postgres=# select securityprofile.suspend_users_noerror ('db_test', 0);
suspend_users_noerror
-----
(1 row)

postgres=# select securityprofile.resume_users_noerror ('db_test', 0);
resume_users_noerror
-----
(1 row)

postgres=#
```

Рисунок 6.24 – Выполнение команды разблокировки пользователей

6.2.1.3 Блокирование/разблокирование группы пользователей администраторов БД

К группе пользователей администраторов БД относятся привилегированные пользователи СУБД, имеющие атрибут CREATEROLE, и возможные атрибуты BYPASSRLS, REPLICATION, а также прочие системные привилегии относительно БД, кроме атрибута CREATEDB.

Компонент «securityprofile» при выполнении SQL-команд по блокировке/разблокировке группы пользователей Администраторы БД отличает их именно по указанным атрибутам. Включение администраторов БД в отдельную групповую роль для работы компонента не требуется.

Блокировка группы учетных записей администраторов БД выполняется тремя функциями:

- suspend_admins – блокировка группы администраторов БД;
- suspend_admins_seconds – блокировка группы администраторов БД с задержкой в секундах;
- suspend_admins_noerror – блокировка группы администраторов БД с игнорированием ошибки ранее заблокированных пользователей.

6.2.1.3.1 Блокировка группы пользователей администраторов БД (suspend_admins)

Функция «suspend_admins» позволяет заблокировать категорию пользователей для заданной БД и имеет синтаксис SQL-команды:

```
suspend_users(bigint, text DEFAULT NULL);
```

Например

Блокировка администраторов БД в БД «postgres» выполняется SQL-командой:

```
SELECT securityprofile.suspend_admins('postgres', 0);
```

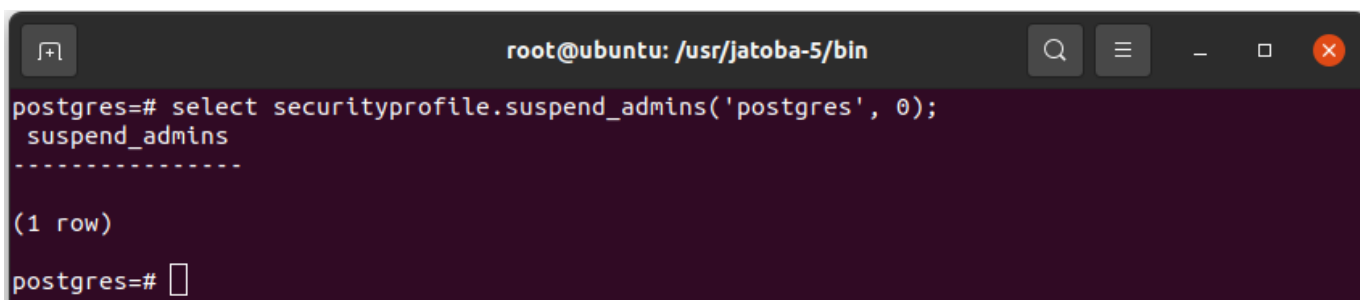


Рисунок 6.25 – Команда блокировки администраторов БД в БД «postgres»

При проверке статуса блокировки, будет добавлена строка с OID БД «5», что соответствует БД «postgres».

```
SELECT * FROM securityprofile.status;  
SELECT oid FROM pg_database WHERE datname = 'postgres';
```

```

root@ubuntu: /usr/jatoba-5/bin

postgres=# SELECT * FROM securityprofile.status;
 databaseoid |      suspendeduserstime       |      suspendedadminstime
-----+-----+-----
          1 | 2023-11-10 03:29:46.37765-08 | 1999-12-31 16:00:00-08
          4 | 2023-11-10 03:29:46.377705-08 | 1999-12-31 16:00:00-08
       16909 | 1976-02-17 02:20:21.631404-08 | 1999-12-31 16:00:00-08
          5 | 2023-11-10 03:29:46.377456-08 | 2023-11-14 07:13:36.095524-08
(4 rows)

postgres=# SELECT oid FROM pg_database WHERE datname = 'postgres';
 oid
-----
    5
(1 row)

postgres=#

```

Рисунок 6.26 – Проверка наличия блокировок

Вывод информации о всех пользователях выполняется SQL-командой:

```
SELECT * from securityprofile.is_locked ('');
```

6.2.1.3.2 Разблокирование группы пользователей администраторов БД (securityprofile.resume_admins)

Разблокировка пользователей выполняется функцией «resume_users», имеющей синтаксис SQL-команды:

```
resume_admins(bigint, text DEFAULT NULL);
```

Например

Разблокируем пользователей БД «db_test» SQL-командой:

```
SELECT securityprofile.resume_admins('postgres', 0);
```

```

root@ubuntu: /usr/jatoba-5/bin

postgres=# select securityprofile.resume_admins('postgres', 0);
 resume_admins
-----
(1 row)

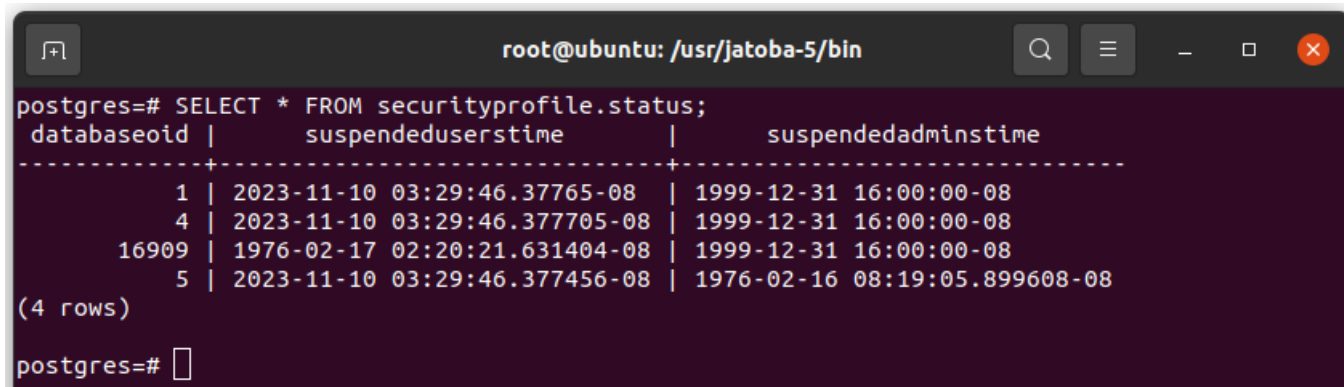
postgres=#

```

Рисунок 6.27 – Разблокирование администраторов БД

Проверка статуса блокировки покажет изменение даты в поле «suspendedadminstime».

```
SELECT * FROM securityprofile.status;
```



```

root@ubuntu: /usr/jatoba-5/bin
postgres=# SELECT * FROM securityprofile.status;
 databaseoid |      suspendeduserstime      |      suspendedadminstime
-----+-----+-----
          1 | 2023-11-10 03:29:46.37765-08 | 1999-12-31 16:00:00-08
          4 | 2023-11-10 03:29:46.377705-08 | 1999-12-31 16:00:00-08
       16909 | 1976-02-17 02:20:21.631404-08 | 1999-12-31 16:00:00-08
          5 | 2023-11-10 03:29:46.377456-08 | 1976-02-16 08:19:05.899608-08
(4 rows)

postgres=#

```

Рисунок 6.28 – Проверка статуса блокировки

6.2.1.3.3 Блокировка группы пользователей администраторов БД с задержкой блокировки в устанавливаемой в секундах (securityprofile.suspend_admins_seconds)

Функция «suspend_admins_seconds» служит для блокировки администраторов БД с задержкой блокировки и имеет синтаксис SQL-команды:

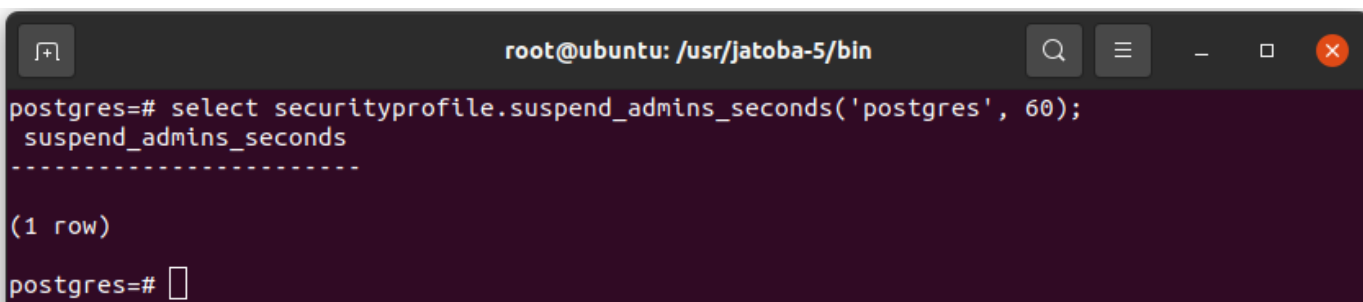
```
suspend_admins_seconds(bigint, text DEFAULT NULL);
```

Например

Блокирование администраторов БД «postgres» с отсрочкой блокировки устанавливаемой в секундах, выполняется SQL-командой:

```
SELECT securityprofile.suspend_admins_seconds('postgres', 60);
```

В SQL-команде указывается БД и время задержки блокировки, устанавливаемой в секундах.



```

root@ubuntu: /usr/jatoba-5/bin
postgres=# select securityprofile.suspend_admins_seconds('postgres', 60);
suspend_admins_seconds
-----
(1 row)

postgres=#

```

Рисунок 6.29 – Блокирование администраторов БД с отсрочкой блокировки устанавливаемой в секундах

6.2.1.3.4 Разблокирование группы пользователей администраторов БД с задержкой блокировки в устанавливаемой в секундах (securityprofile.resume_admins_seconds)

Разблокирование администраторов БД выполняется функцией «resume_admins_seconds» имеющей синтаксис:

```
resume_admins_seconds(bigint, text DEFAULT NULL);
```

Например

Разблокируем администраторов БД «postgres» с задержкой в секундах, SQL-командой:

```
SELECT securityprofile.resume_admins_seconds('postgres', 0);
```

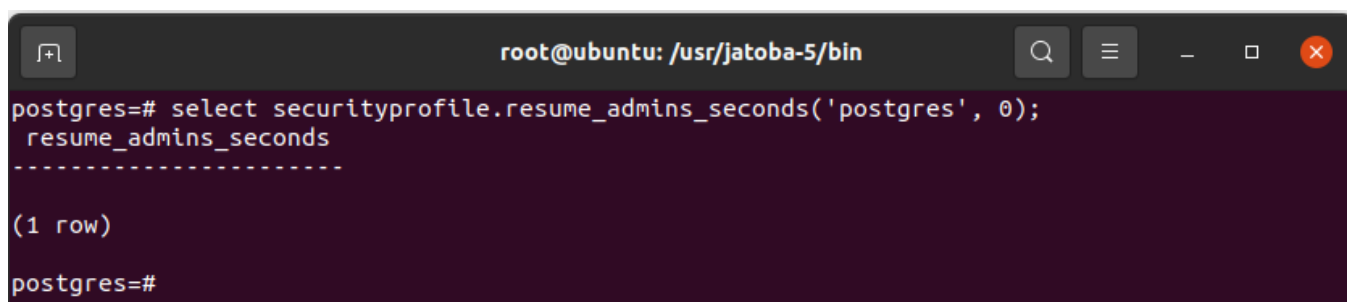


Рисунок 6.30 – Команда разблокировки администраторов БД с установленной задержкой по времени

6.2.1.3.5 Блокировка группы пользователей администраторов БД с игнорированием ошибки (securityprofile.admins_users_noerror)

Функция «suspend_admins_noerror» аналогично «suspend_admins», только не выдает ошибки при наличии уже установленной блокировки и имеет синтаксис SQL-команды:

```
suspend_admins_noerror(text, bigint);
```

Например

Блокировка администраторов БД «postgres» игнорированием ошибки, выполняется SQL-командой:

```
SELECT securityprofile.suspend_admins_noerror ('postgres', 0);
```

```

root@ubuntu: /usr/jatoba-5/bin
postgres=# select securityprofile.suspend_admins_noerror ('postgres', 0);
suspend_admins_noerror
-----
(1 row)
postgres=#
  
```

Рисунок 6.31 – Блокировка администраторов БД «postgres» игнорированием ошибки

6.2.1.3.6 Разблокировка группы пользователей администраторов БД с игнорированием ошибки (resume_admins_noerror)

Разблокирование администраторов БД выполняется функцией «resume_admins_noerror» имеющей синтаксис:

```
resume_admins_noerror(text, bigint);
```

Например

Разблокируем администраторов БД SQL-командой вне зависимости от имеющихся ошибок:

```
SELECT securityprofile.resume_admins_seconds('db_test', 0);
```

```

root@ubuntu: /usr/jatoba-5/bin
postgres=# select securityprofile.resume_admins_seconds('db_test', 0);
resume_admins_seconds
-----
(1 row)
postgres=#
  
```

Рисунок 6.32 – Выполнение команды разблокировки пользователей

6.2.2. Проверка установленных блокировок

Компонент «securityprofile» обладает функциональной возможностью проверки установленных блокировок учетных записей пользователей функциями:

Проверить наличие блокировок в БД возможно следующими функциями:

- is_users_suspended – блокировка пользователей (см. п. 6.2.2.1);
- is_admins_suspended – блокировка администраторов БД (см. п. 6.2.2.2);

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

- is_suspended – существование блокировки (см. п. 6.2.2.3).

Если в любой базе существует блокировка категории пользователя, то возвращаем «true», при отсутствии блокировкой выводится значение «false».

6.2.2.1 Проверка блокировки пользователей

Проверка блокировки группы пользователей выполняется SQL-командой, имеющей синтаксис:

```
SELECT securityprofile.is_users_suspended (text DEFAULT NULL);
```

Например

Сформируем SQL-команду для вывода наличия блокировки пользователей:

```
SELECT securityprofile.is_users_suspended ('db_test');
```

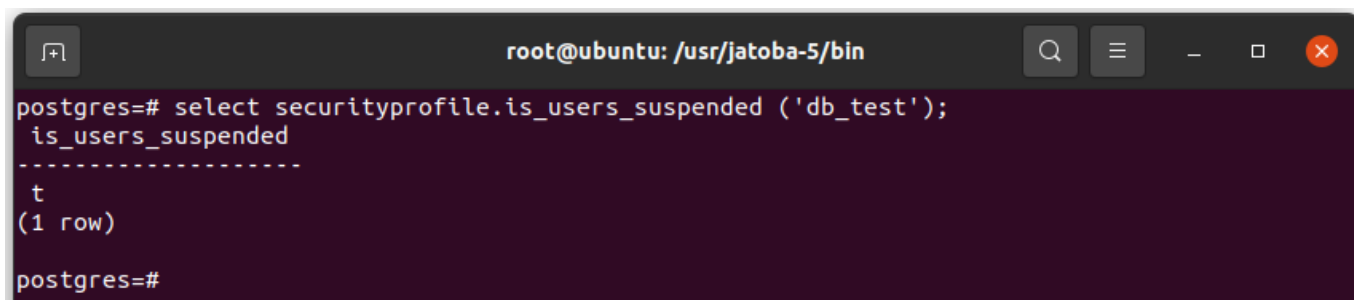


Рисунок 6.33 – Вывод состояния блокировки пользователей в БД

6.2.2.2 Проверка блокировки группы администраторов БД

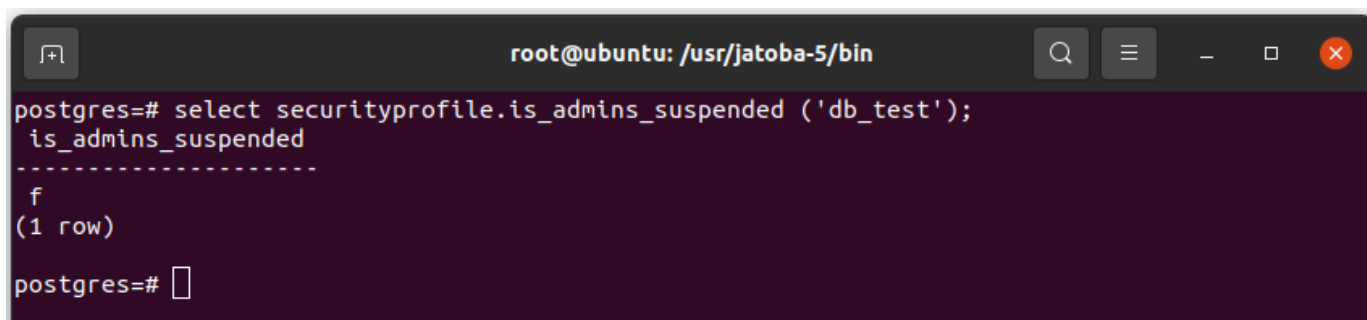
Проверка блокировки группы администраторов БД выполняется SQL-командой, имеющей синтаксис:

```
SELECT securityprofile.is_admins_suspended (text DEFAULT NULL);
```

Например

Сформируем SQL-команду для вывода наличия блокировки администраторов БД:

```
SELECT securityprofile.is_admins_suspended ('db_test');
```

```

root@ubuntu: /usr/jatoba-5/bin
postgres=# select securityprofile.is_admins_suspended ('db_test');
 is_admins_suspended
-----
 f
(1 row)
postgres=#

```

Рисунок 6.34 – Вывод состояния блокировки администраторов БД

6.2.2.3 Проверка существования блокировки

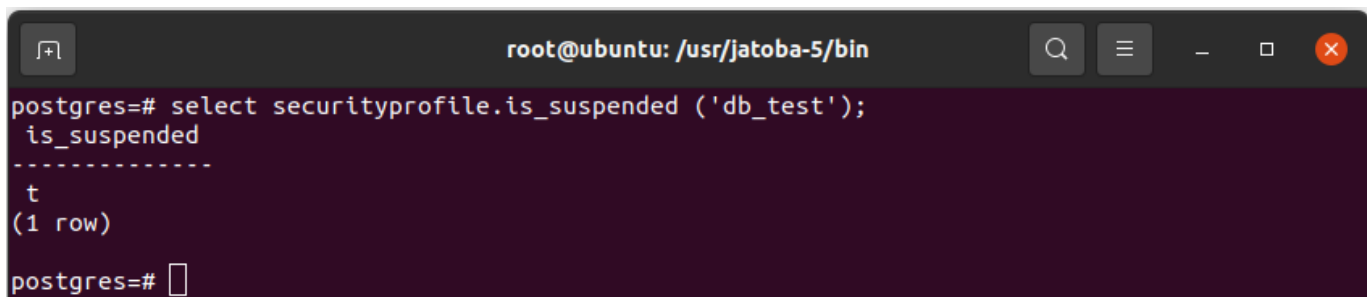
Проверка блокировки любой группы пользователей СУБД, как группы администраторов БД, так и группы пользователей, выполняется SQL-командой, имеющей синтаксис:

```
SELECT securityprofile.is_suspended (text DEFAULT NULL);
```

Например

Сформируем SQL-команду для вывода наличия блокировок:

```
SELECT securityprofile.is_suspended ('db_test');
```



```

root@ubuntu: /usr/jatoba-5/bin
postgres=# select securityprofile.is_suspended ('db_test');
 is_suspended
-----
 t
(1 row)
postgres=#

```

Рисунок 6.35 – Вывод состояния блокировок в БД

6.2.3. Создание новых ролей, присвоение атрибутов и системных привилегий

СУБД «Jatoba» поддерживает создание новых ролей (пользователей) с атрибутами ролей, приведенных в таблице 6.12. Для назначения атрибутов ролям необходимо выполнить следующую команду:

```
ALTER ROLE <имя учетной записи пользователя> with <атрибут ролей из таблицы 6.12>;
```

Таблица 6.12 – Атрибуты ролей

Атрибут	Условный перевод	Описание
SUPERUSER	Суперпользователь	Роль «Суперпользователь» обладает полными правами доступа к СУБД
INHERIT	Наследование	Роли, имеющие атрибут «INHERIT», автоматически используют права всех ролей, членами которых они являются, в том числе и унаследованные этими ролями права.
CREATEROLE	Право создание роли	Роль имеет разрешение на создание других ролей. Роль с правом «CREATEROLE» может не только создавать, но и изменять и удалять другие роли, а также выдавать и отзывать членство в ролях
CREATEDB	Право создания базы данных	Роль имеет разрешение на создание базы данных.
LOGIN	Право входа	Роль с атрибутом «LOGIN» рассматривается, как роль пользователя базы данных, а также может использоваться для начального подключения к базе данных.
REPLICATION	Право репликации	Роль имеет разрешение на запуск потоковой репликации.
BypassRls		Атрибут роли, определяющий игнорирование все политики защиты на уровне строк (RLS – Row Level Security)

СУБД «Jatoba» регулирует системные привилегии для ролей с атрибутом «Login» в соответствии с таблицей 6.13.

Таблица 6.13 – Системные привилегии для ролей с атрибутом «LOGIN»

Наименование	Синтаксис предоставления разрешений роли на конкретные объекты	Описание
SELECT LARGE OBJECT	GRANT { { SELECT UPDATE } [, ...] ALL [PRIVILEGES] }	Получение больших объектов
UPDATE LARGE OBJECT	ON LARGE OBJECT <идентификатор большого объекта> [, ...] TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Изменение данных в больших объектах
SELECT SEQUENCE	GRANT { { USAGE SELECT UPDATE } [, ...] ALL [PRIVILEGES] }	Получение значения последовательностей
UPDATE SEQUENCE	ON { SEQUENCE <название последовательности> [, ...]	Обновление значения последовательностей
USAGE SEQUENCE	ALL SEQUENCES IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Использование последовательности

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

Наименование	Синтаксис предоставления разрешений роли на конкретные объекты	Описание
TRIGGER TABLE	GRANT { { TRIGGER } [, ...] ALL [PRIVILEGES] } ON { [TABLE] <название таблицы> [, ...] ALL TABLES IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Установка триггеров на таблицы
REFERENCES TABLE	GRANT { { REFERENCES } [, ...] ALL [PRIVILEGES] } ON { [TABLE] <название таблицы> [, ...] ALL TABLES IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Использование зависимых таблиц
REFERENCES TABLE COLUMN	GRANT { { REFERENCES } (<название столбца> [, ...]) [, ...] ALL [PRIVILEGES] (<название столбца> [, ...]) } ON [TABLE] <название таблицы> [, ...] TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Использование колонок зависимых таблиц
CREATE DATABASE	GRANT { { CREATE CONNECT TEMPORARY } [, ...] ALL [PRIVILEGES] } ON DATABASE <название базы данных> [, ...] TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Создание базы данных
CONNECT DATABASE		Подключение к базе данных
TEMPORARY DATABASE		Использование временных баз данных
CREATE SCHEMA	GRANT { { CREATE USAGE } [, ...] ALL [PRIVILEGES] } ON SCHEMA <название схемы> [, ...] TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Создание схемы
USAGE SCHEMA		Использование схемы
CREATE TABLESPACE	GRANT { CREATE ALL [PRIVILEGES] } ON TABLESPACE <название табличного пространства> [, ...] TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Создание табличного пространства

Наименование	Синтаксис предоставления разрешений роли на конкретные объекты	Описание
EXECUTE FUNCTION	GRANT { EXECUTE ALL [PRIVILEGES] } ON { FUNCTION <название функции> ([[<режим аргумента>] [<название аргумента>] <тип данных аргументов функции> [, ...]) [, ...] ALL FUNCTIONS IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Выполнение функций
USAGE DOMAIN	GRANT { USAGE ALL [PRIVILEGES] } ON DOMAIN <название домена> [, ...] TO <название роли> [, ...] [WITH GRANT OPTION]	Использование домена
USAGE FOREIGN DATA WRAPPER	GRANT { USAGE ALL [PRIVILEGES] } ON FOREIGN DATA WRAPPER <название внешнего источника данных> [, ...] TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Использование внешних источников данных
USAGE FOREIGN SERVER	GRANT { USAGE ALL [PRIVILEGES] } ON FOREIGN SERVER <название сервера> [, ...] TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Использование внешних серверов
USAGE LANGUAGE	GRANT { USAGE ALL [PRIVILEGES] } ON LANGUAGE <название языка программирования> [, ...] TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Использования языка программирования
USAGE TYPE	GRANT { USAGE ALL [PRIVILEGES] } ON TYPE <название типа> [, ...] TO <название роли> [, ...] [WITH GRANT OPTION]	Использование тип

СУБД «Jatoba» поддерживает права субъекта доступа, указанные в таблице 6.14, к объектам доступа. Для предоставления привилегий необходимо использовать инструкции «GRANT».

Таблица 6.14 – Права субъекта доступа к объектам доступа

Право	Синтаксис	Описание
SELECT	<p>на таблицу: GRANT { { SELECT } [, ...] ALL [PRIVILEGES] } ON { [TABLE] <название таблицы> [, ...] ALL TABLES IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]</p> <p>на столбец: GRANT { { SELECT } [, ...] ALL [PRIVILEGES] } ON { [TABLE] <название таблицы> [, ...] ALL TABLES IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]</p>	Позволяет читать содержание объекта, а также выполнять команду «SELECT» для любого столбца или перечисленных столбцов в заданной таблице, представлении или последовательности
INSERT	GRANT { { INSERT } [, ...] ALL [PRIVILEGES] } ON { [TABLE] <название таблицы> [, ...] ALL TABLES IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Позволяет вставлять строки в заданную таблицу с помощью команды «Insert»
UPDATE	<p>на таблицу: GRANT { { UPDATE } [, ...] ALL [PRIVILEGES] } ON { [TABLE] <название таблицы> [, ...] ALL TABLES IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]</p> <p>на столбец: GRANT { { UPDATE } [, ...] ALL [PRIVILEGES] } ON { [TABLE] <название таблицы> [, ...] ALL TABLES IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]</p>	Позволяет изменять с помощью команды «UPDATE» данные во всех либо только перечисленных столбцах в заданной таблице
DELETE	GRANT { { DELETE } [, ...] ALL [PRIVILEGES] } ON { [TABLE] <название таблицы> [, ...] ALL TABLES IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Позволяет удалять строки из заданной таблицы с помощью команды «DELETE»
№ изменения: _____ Подпись отв. лица: _____ Дата внесения изм: _____		

Право	Синтаксис	Описание
TRUNCATE	GRANT { { TRUNCATE } [, ...] ALL [PRIVILEGES] } ON { [TABLE] <название таблицы> [, ...] ALL TABLES IN SCHEMA <название схемы> [, ...] } TO { [GROUP] <название роли> PUBLIC } [, ...] [WITH GRANT OPTION]	Позволяет опустошить заданную таблицу или набор таблиц с помощью команды «TRUNCATE»

6.2.4. Создание ролей при активированной парольной политике

Создание пользователей при активированной парольной политике доступно только при условии подключения к БД, в которой установлено расширение SecurityProfile (см. п. 6.1.3). Администратор БД должен выполнить подключение к такой БД перед выполнением дальнейших действий.

Создание пользователей при активированной парольной политике выполняется двумя способами:

- Без указания пароля:

```
CREATE USER <имя учетной записи пользователя>;
```

Пользователь, для которого не установлен пароль, не может выполнять подключение к БД. Процедура установки пароля, при активированной парольной политике, приведена в . п. 6.2.5 данного руководства.

- С указанием паролем:

```
CREATE USER <имя учетной записи пользователя> PASSWORD '<пароль пользователя>';
```

6.2.5. Установка пароля пользователя

После создания пользователя, с учетом включенной парольной политики, его пароль возможно изменить одним из двух способов:

- Установка пароля пользователя, в случае если он не был указан при создании пользователя:

```
ALTER ROLE <имя учетной записи пользователя> PASSWORD '<пароль пользователя>';
```

- Изменение существующего пароля пользователя в интерактивном режиме:

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

```
\password <имя_пользователя>
Enter new password for user <имя_пользователя>": <пароль
пользователя>
Enter it again: <пароль пользователя>
```



Применение команды `\password` возможно только при отключении запрета на обработку хэшированных паролей компонентом SecurityProfile. Для этого необходимо отключить параметр `securityprofile.hashed_password_strictness` в конфигурационном файле `postgres.conf`:

```
securityprofile.hashed_password_strictness = off
```

После внесения изменений в конфигурационный файл необходимо выполнить перезагрузку сервиса СУБД.

6.2.6. Блокирование сеанса доступа в СУБД после установленного времени бездействия (неактивности) пользователя

Блокирование сеанса доступа в СУБД после установленного времени бездействия (неактивности) пользователя выполняется установкой параметров:

- `idle_session_timeout`;
- `idle_in_transaction_session_timeout`.

Параметры устанавливаются в разделе «CLIENT CONNECTION DEFAULTS» конфигурационного файла `postgresql.conf`.



Рисунок 6.36 – Параметры «`idle_session_timeout`» и «`idle_in_transaction_session_timeout`» в конфигурационном файле `postgresql.conf`

```
idle_session_timeout
```

Параметр «`idle_session_timeout`» в СУБД определяет время (в секундах), в течение которого неактивная сессия будет оставаться открытой. По истечении этого времени неактивная сессия закрывается и все связанные с ней ресурсы освобождаются. Этот

параметр может быть полезен для управления ресурсами сервера и предотвращения атак, связанных с захватом соединений (connection hijacking attacks).

```
idle_in_transaction_session_timeout
```

В СУБД параметр «idle_in_transaction_session_timeout» определяет время неактивности (в секундах) для сессии, которая участвует в транзакции. По истечении этого времени такая сессия будет автоматически завершена.

Это помогает предотвратить ситуации, когда сессии «застревают» в состоянии ожидания из-за долго выполняющихся транзакций. Это особенно важно, если есть длительные транзакции, которые могут блокировать другие запросы к БД.

По умолчанию значение этого параметра равно 0, что означает отсутствие ограничения на время неактивности. Однако, рекомендуется установить значение, которое подходит для используемого приложения и среды.

6.2.7. Прерывание текущих сессий в БД

Функция securityprofile.terminate_backend(database_name text), используется для прерывания всех активных сессии, всех категорий пользователей в указанной базе данных.

Например

```
SELECT securityprofile.terminate_backend('db_test');
```

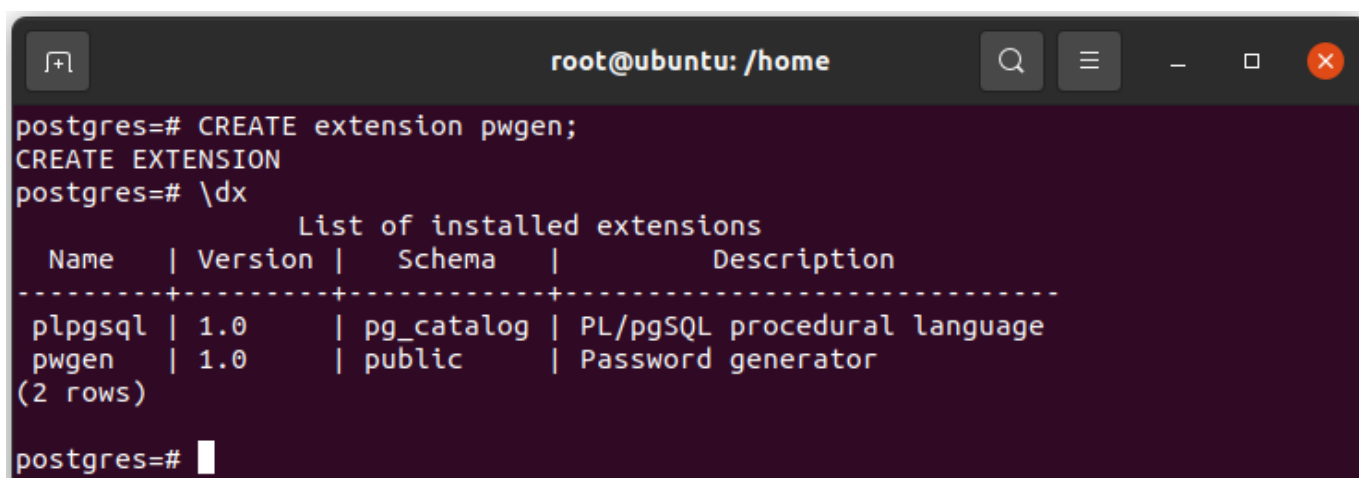

6.3. Генератор паролей

6.3.1. Установка расширения pwgen

Установка компонента pwgen не требует установки пакета, т.к. исходный файл расширения устанавливается при базовой установке СУБД.

Расширение устанавливается от имени и с правами привилегированного пользователя SQL-командой:

```
# CREATE extension pwgen;  
# \dx
```



```
root@ubuntu: /home  
postgres=# CREATE extension pwgen;  
CREATE EXTENSION  
postgres=# \dx  
  
List of installed extensions  
Name | Version | Schema | Description  
-----+-----+-----+-----  
plpgsql | 1.0 | pg_catalog | PL/pgSQL procedural language  
pwgen | 1.0 | public | Password generator  
(2 rows)  
postgres=#
```

Рисунок 6.37 – Установка расширения «pwgen»

Расширение работает только в той базе, в которой его установил администратор.

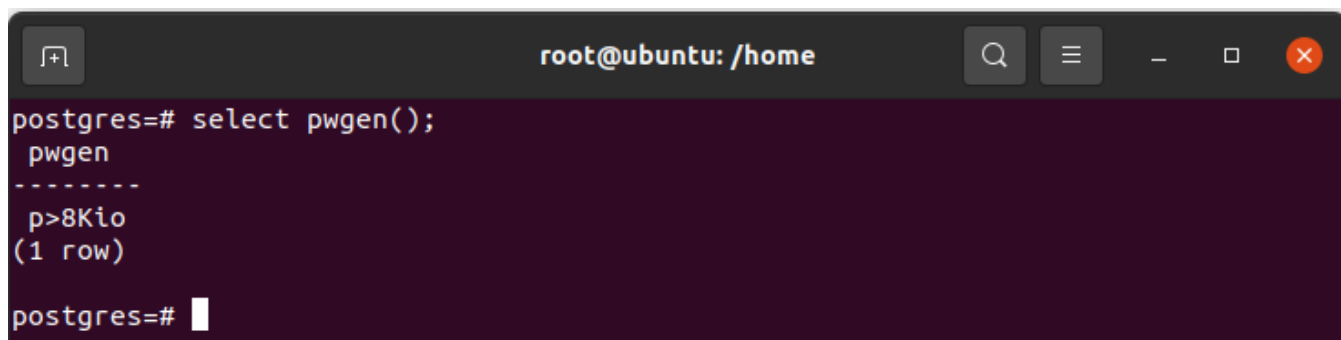
6.3.2. Генерация пароля

Вызов функции генерации пароля доступен для всех пользователей СУБД.

Генерация паролей осуществляется вызовом специальной SQL-функции. Функция имеет несколько параметров, через которые можно определять состав символов в генерируемых паролях. По правилам SQL значения параметров функции можно задать либо последовательно друг за другом, либо указав конкретное название параметра функции и его значение.

Без аргументов, по умолчанию, расширение сгенерирует шестисимвольный пароль с одним спецсимволом:

```
SELECT pwgen ( ) ;
```



```
root@ubuntu: /home
postgres=# select pwgen();
pwgen
-----
p>8Kio
(1 row)

postgres=#
```

Рисунок 6.38 – Генерация пароля с параметрами по умолчанию
SQL-команда с указанием последовательности значений имеет синтаксис:

```
SELECT pwgen (
[длина пароля] ,
[количество строчных букв] ,
[количество прописных букв] ,
[количество цифр] ,
[количество специальных символов] ,
['допустимый набор строчных символов'] ,
['допустимый набор прописных символов'] ,
['допустимый набор специальных символов'] ) ;
```

Например

SQL-команда генерации пароля со следующими характеристиками:

- длиной 10 символов;
- 2 строчными буквами;
- 2 прописными буквами;
- 1 цифрой;
- 1 специальным символом;
- допустимый набором строчных символов - 'abcde';
- допустимым набором прописных символов - 'ABCDE';
- допустимым набором специальных символов - '@!';

будет следующей:

```
# SELECT pwgen(10,2,2,1,1,'abcde','ABCDE','@!');
```

```
root@ubuntu: /home
postgres=# select pwgen(10,2,2,1,1,'abcde','ABCDE','@!');
pwgen
-----
Ee@B'eb@A#
(1 row)
```

Рисунок 6.39 – SQL-команда генерации пароля по последовательности параметров

Количество задаваемых букв не должно превышать длину пароля.

Значение 0 в параметрах количества символов функции pwgen означает, что не накладывается никаких ограничений на присутствие символов из соответствующего алфавита, заданного параметрами 6-10.

Параметры, задающие количество символов в пароле, определяют минимальное число символов в пароле каждого типа. Возможен вывод большего числа символов.

Сгенерировать пароли по задаваемым аргументам возможно используя параметры приведенными в таблице 6.15.

Таблица 6.15 – Параметры и значения для генерации пароля

Параметры	Описание	Значения
pw_len	длина генерируемого пароля	по умолчанию 6
pw_lc_char_cnt	количество строчных (low case) символов в пароле	по умолчанию минимум 1
pw_uc_char_cnt	количество прописных (upper case) символов в пароле	по умолчанию минимум 1
pw_num_cnt	количество цифр в пароле	по умолчанию минимум 1
pw_spec_cnt	количество специальных символов в пароле	по умолчанию минимум 1
pw_lc_char_allowed	допустимый набор строчных символов, из которых генерируется пароль	'abcdefghijklmnopqrstuvwxyz'
pw_uc_char_allowed	допустимый набор прописных символов, из которых генерируется пароль	'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
pw_spec_allowed	допустимый набор специальных символов, из которых генерируется пароль	'\!"#\$%&()*+,-./:;<=>?@[]^_`{ }~'

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

Например

SQL-команда генерации пароля со следующими характеристиками:

- длиной 10 символов (pw_len =>10);
- 3 строчными символами в пароле (pw_lc_char_cnt =>3);
- 2 цифрами в пароле (pw_num_cnt => 5).

```
# SELECT pwgen(pw_len =>10, pw_lc_char_cnt =>3, pw_num_cnt => 5);
```

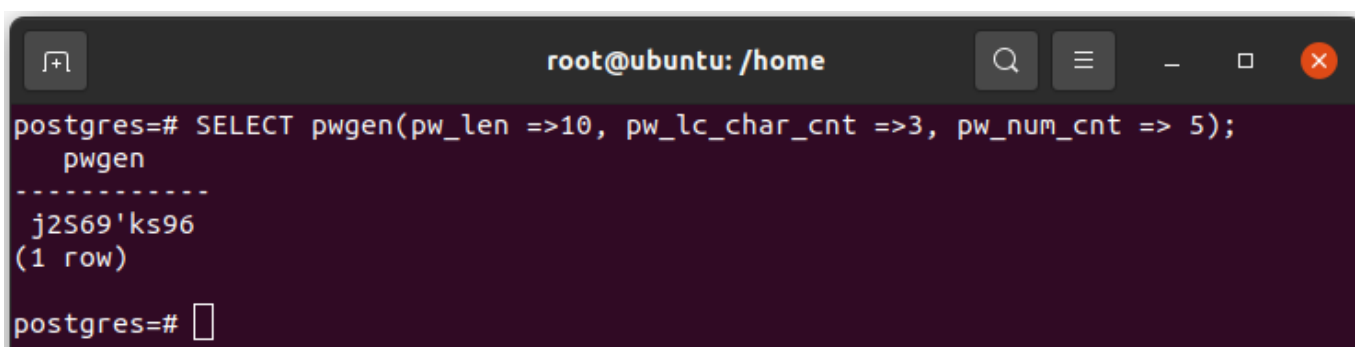


Рисунок 6.40 – Генерация пароля по заданным параметрам

6.3.3. Генерация множества паролей

Расширение «pwgen» для генерации множества паролей использует функцию СУБД возвращения множества. В которой задается начало отсчета и конец отсчета.

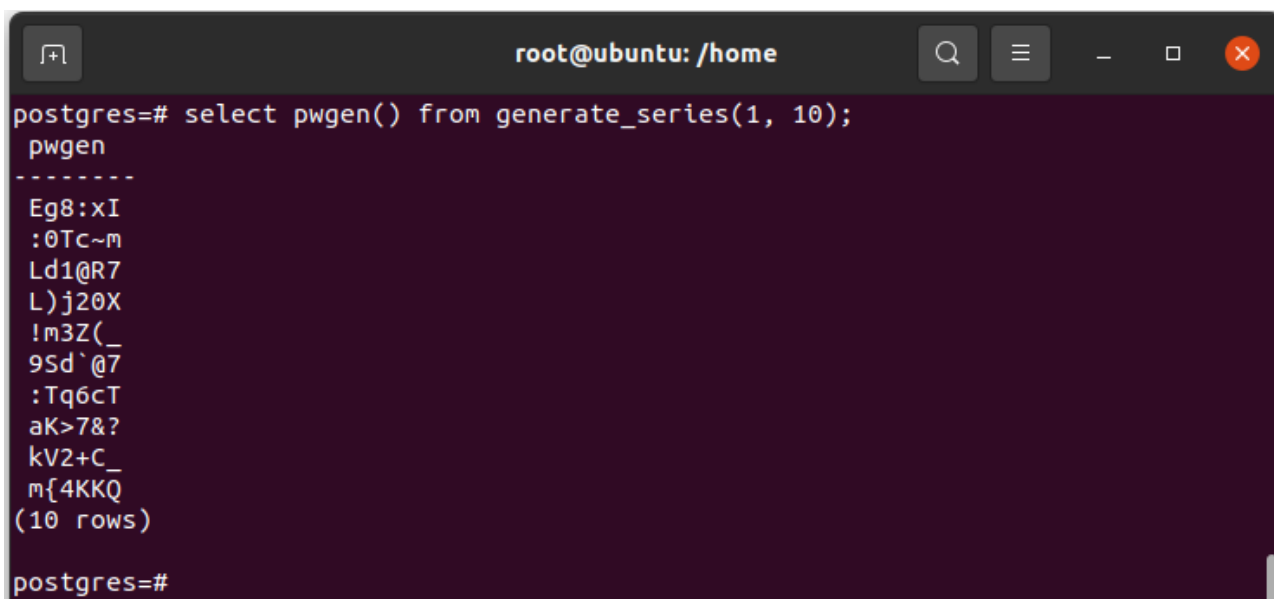
SQL-команда имеет синтаксис:

```
SELECT pwgen() from generate_series(start integer, stop integer);
```

Например

SQL-команда генерации множества паролей с началом отсчета от единицы и концом отсчета до 10 будет следующей:

```
select pwgen() from generate_series(1, 10);
```



```

root@ubuntu: /home
postgres=# select pwgen() from generate_series(1, 10);
 pwgen
-----
Eg8:xI
:0Tc~m
Ld1@R7
L)j20X
!m3Z(_
9Sd`@7
:Tq6cT
aK>7&?
kV2+C_
m{4KKQ
(10 rows)

postgres=#
  
```

Рисунок 6.41 – Генерация множества паролей

6.3.4. Удаление расширения «pwgen»

Расширение удаляется SQL-командой:

```
DROP extension pwgen;
```

6.4. Регистрация событий безопасности СУБД «Jatoba»

6.4.1. Настройки регистраций событий безопасности СУБД «Jatoba» под управлением ОС Windows Server

Функциональная возможность СУБД, а именно функция syslogcollector (внутренний механизм), позволяет отправлять события в хранилище ОС Windows Server.

Для настройки регистрации событий безопасности СУБД «Jatoba» необходимо выполнить ряд действий:

- 1) От учетной записи администратора СУБД подключиться к ОС и открыть файл «postgresql.conf»⁴).
- 2) В файле «postgresql.conf» прописать следующие параметры:

```

log_destination = 'eventlog'
logging_collector = on
log_connections = on
  
```

⁴ Местонахождение файла «postgresql.conf» в ОС Windows Server: C:\Program Files\GIS\Jatoba\<ver>\data\postgresql.conf

```
log_disconnections = on
log_statement = 'mod'
log_hostname = on
```

3) Перезагрузить СУБД «Jatoba» при помощи команды:

```
net stop JatobaServer
net start JatobaServer
```

Расшифровка параметров по регистрации событий СУБД «Jatoba», которая установлена в ОС Windows Server, представлены в таблице 6.16.

Таблица 6.16 – Расшифровка параметров по регистрации событий СУБД «Jatoba»

Параметр	Описание
log_destination	Данный параметр указывает на то, где будут храниться события СУБД «Jatoba»
logging_collector	Данный параметр позволяет включать сборщик журналов, который перенаправляет сообщения в файл журнала
log_connections	Включение данного параметра позволяет регистрировать все подключения к СУБД «Jatoba», включая неуспешные
log_disconnections	Включение данного параметра позволяет регистрировать завершение сеанса
log_statement	Данный параметр управляет, тем какие SQL-команды будут регистрироваться. Параметр mod позволяет записывать следующие команды: все команды DDL (CREATE, ALTER, DROP), а так же INSERT, UPDATE, DELETE, TRUNCATE и COPY FROM. PREPARE, EXECUTE и EXPLAIN ANALYZE
log_hostname	Включение данного параметра позволяет регистрировать имя хоста

6.4.2. Настройки регистрации событий безопасности СУБД «Jatoba» под управлением ОС семейства GNU/Linux

Функциональная возможность СУБД, а именно функция syslogcollector (внутренний механизм), позволяет отправлять события из папки в хранилище ОС семейства GNU/Linux.

Для настройки регистрации событий безопасности СУБД «Jatoba» необходимо выполнить ряд действий:

1) От учетной записи администратора СУБД подключиться к ОС и открыть файл «postgresql.conf»⁵⁾.

⁵) Местонахождение файла «postgresql.conf» в ОС семейства GNU/Linux: /var/lib/jatoba/<ver>/data/postgresql.conf

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

2) В файле «postgresql.conf» перепроверить следующие параметры:

```
#-----
# JATOBA LOGGING PARAMETERS
#-----
log_destination = 'stderr'
logging_collector = on
log_directory = 'log'
log_filename = 'jatoba-%a.log'
log_rotation_age = 1d
log_rotation_size = 0
log_truncate_on_rotation = on
log_line_prefix = '%m [%p] '
```

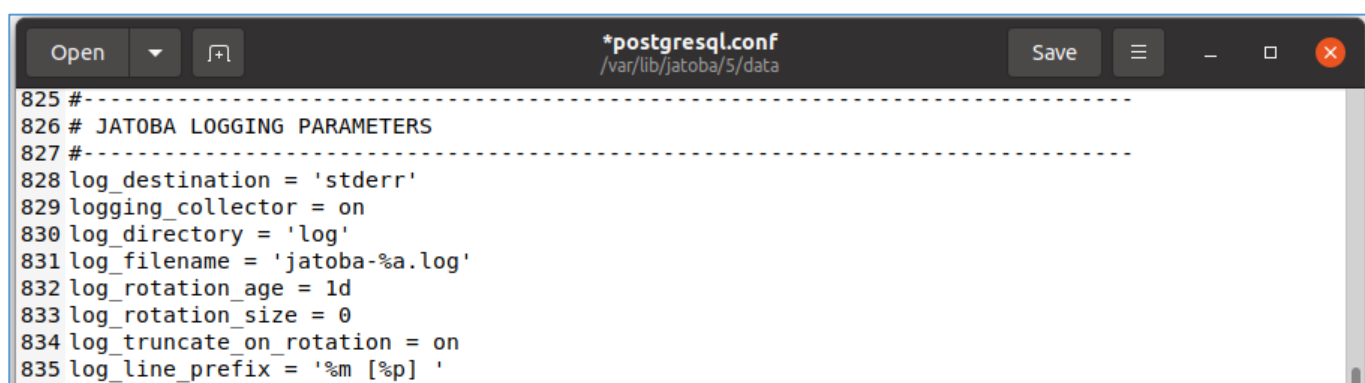


Рисунок 6.42 – Предустановленные параметры логирования



Если не планируется использовать компоненты pgAudit и pgauditlogtofile, то будет целесообразным установить параметр регистрации событий СУБД log_statement.

3) Перезагрузить СУБД «Jatoba» при помощи команды, если параметры были изменены:

```
# systemctl restart jatoba-<ver>
```

Параметры регистрации событий и их значения, используемые СУБД и ее компонентами представлены в таблице 6.17.

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

Таблица 6.17 – Параметры и их значения СУБД и ее компонентов

Параметр	Параметр СУБД	Параметр компонента	Значение	Описание	Компонент	shared_preload_libraries	EXTENSION
log_destination	X	—	'csvlog'	Данный параметр указывает на то, где будут храняться события СУБД «Jatoba»	ja_csum	X	—
	X	—	'csvlog'		ja_Log	—	—
	X	—	'stderr,csvlog'		auto_explain	X	X
log_connections	X	—	on	Параметр позволяет регистрировать все подключения к СУБД «Jatoba», включая неуспешные	ja_csum	X	—
					pgBadger	—	—
log_disconnections	X	—	ON	Включение данного параметра позволяет регистрировать завершения сеанс	pgBadger	—	—
log_directory	X	—	'log'	Параметр указывает на директорию хранения журнала аудита СУБД	ja_csum	X	—
					auto_explain	X	X
					ja_Log	—	—
log_min_messages	X	—	info	Параметр определяющий минимальный уровень записываемых сообщений	ja_csum	X	—
log_filename	X	—	log_filename = 'jatoba-%Y-%m-%d_%H%M%S.log'	Параметр задаёт имя файла журнала аудита СУБД	auto_explain	X	X
					ja_Log	—	—
log_rotation_age	X	—	1D	Параметр определяет максимальное время	auto_explain	X	X
№ изменения: _____		Подпись отв. лица: _____		Дата внесения изм: _____			

				жизни файла аудита событий СУБД			
log_rotation_size	X	—	0	Параметр определяет ротацию файла аудита СУБД по размеру. По умолчанию значение -10 Мб. При нулевом значении ротация по размеру файла не производится.	auto_explain	X	X
log_truncate_on_rotation	X	—	ON	Параметр устанавливает перезапись существующего файла аудита СУБД, а не дополнительную запись в них.	auto_explain	X	X
log_line_prefix	X	—	'%t [%p]: user=%u,db=%d, app=%a,client=% h'	Параметр устанавливает наличие статусной информации в строках журнала аудита СУБД	pgBadger	—	—
			'%m [%p] app=%a host=%h user=%u db=%d '		auto_explain	X	X
syslog_facility	X	—	LOCAL0	Параметр используется при включенном при использовании syslog			
syslog_ident	X	—	'postgres'	При использовании syslog данный параметр указывает имя программы, используемое для идентификации сообщений			

log_timezone	X	—		Параметр указывает на часовой пояс, используемых для отметок времени, записываемых в журнале			
log_mask_password	X	—	1	Параметр включающий функцию маскирования паролей	log_mask_password	—	—
log_statement	X	—	all	Данный параметр управляет, тем какие SQL-команды будут регистрироваться.	ja_Log	—	—
					log_mask_password	—	—
log_hostname	X	—	on	Включение данного параметра позволяет регистрировать имя хоста			
track_io_timing	X	—	on	Параметр включает мониторинг времени чтения и записи блоков	auto_explain	X	X
log_min_duration_statement	X	—	10	Параметр записывает продолжительность выполнения всех команд, время работы которых не меньше указанного	pgBadger	—	—
					auto_explain	X	X
					log_mask_password	—	—
log_duration	X	—	true	Параметр в СУБД, который контролирует, будет ли длительность выполненных запросов регистрироваться в журнале. Если log_duration включён (установлен в true), СУБД будет регистрировать длительность каждого	log_mask_password		

				выполненного SQL-запроса в журнале сервера.			
log_checkpoints	X	—	on	Параметр включает регистрацию выполнения контрольных точек и точек перезапуска сервера	pgBadger	—	—
log_lock_waits	X	—	on	Параметр определяет, нужно ли фиксировать в журнале события, когда сеанс ожидает получения блокировки дольше, чем указано в deadlock_timeout	pgBadger	—	—
log_temp_files	X	—	0	Параметр включает регистрацию имен и размеров временных файлов	pgBadger	—	—
log_autovacuum_min_duration	X	—	0	Параметр включает регистрацию действий по автоочистке	pgBadger	—	—
log_error_verbosity	X	—	default	Параметр определяет количество детальной информации	pgBadger	—	—
logging_collector	X	—	on	Параметр позволяет включать сборщик журналов, который перенаправляет сообщения в файл журнала	auto_explain	X	X
					ja_csum	X	—
					ja_Log	—	—
auto_explain.log_min_duration	—	X	10	минимальное время выполнения запроса в миллисекундах, при превышении которого	auto_explain	X	X

				план запроса будет попадать в лог			
auto_explain.log_nested_statements	—	X	true	Параметр включает регистрацию планов выполнения вложенных операторов (операторов, выполняемых внутри функции)	auto_explain	X	X
auto_explain.log_analyze	—	X	true	Параметр определяет регистрацию плана выполнения запроса	auto_explain	X	X
auto_explain.log_buffers	—	X	true	Параметр определяет, будет ли регистрироваться статистика использования буферного кеша	auto_explain	X	X
auto_explain.log_triggers	—	X	on	Параметр определяет регистрацию выполнения триггеров	auto_explain	X	X
track_io_timing	X	—	'on'	Параметр включает мониторинг времени чтения и записи блоков	auto_explain	X	X

6.4.3. Компонент «pgAudit». Настройка расширенной регистрации событий безопасности

Компонент «pgAudit» обеспечивает расширенное журналирование событий. Компонент выполнен в виде расширения СУБД.

Версия компонента:

- с версией ядра «4» - 1.6.2;
- с версией ядра «5» - 1.7.0;
- с версией ядра «6» - 16.0.0;

Установка пакета компонента описана в документе 643.72410666.00067-07 97 01 «Руководство по установке».

После установки компонента поле «error message» расширяется данными:

- тип записи события;
- № выражения;
- № подвыражения;
- класс события;
- SQL–операция;
- тип объекта БД;
- имя объекта БД;
- полный текст SQL–запроса (скрипта);
- параметры SQL–запроса (скрипта).

Структура полей событий безопасности меняется, как представлено в таблице 6.18.

Таблица 6.18 – Структура полей событий безопасности

№	Стандартные поля событий безопасности	
		Поля событий безопасности с pgAudit
1	Time stamp with milliseconds	Штамп времени с миллисекундами
2	Criticality	Критичность события

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

№	Стандартные поля событий безопасности		
		Поля событий безопасности с pgAudit	
3	Class		Тип события
4	User name		Имя пользователя
5	Database name		Имя базы данных
6	Process ID		Идентификатор процесса
7	Client host		Клиентский узел
8	Port number		Номер порта
9	Session ID		Идентификатор сессии
10	Per-session line numbe		Номер строки каждой сессии
11	Command tag		Тег команды
12	Session start time		Время начала сессии
13	Virtual transaction ID		Виртуальный идентификатор транзакции
14	Regular transaction ID		Идентификатор транзакции
15	Error severity		Уровень важности ошибки
16	SQLSTATE code		Код ошибки SQLSTATE
17	Error message		Сообщение об ошибке
17.1		Audit_type	Тип записи события
17.2		Statement_id	№ выражения
17.3		Substatement_id	№ подвыражения
17.4		Class	класс события
17.5		Command	SQL-операция
17.6		Object_type	Тип объекта БД
17.7		Object_name	Имя объекта БД
17.8		Statement	Полный текст SQL-запроса (скрипта)
17.9		Parameter	Параметры SQL-запроса (скрипта)
18	Error message detail		Подробности к сообщению об ошибке
19	Hint		Подсказка к сообщению об ошибке
20	Internal query that led to the error		Внутренний запрос
21	Character count of the error position therein		Номер символа внутреннего запроса, где произошла ошибка
22	Error context		Контекст ошибки
23	User query that led to the error		Запрос пользователя

№	Стандартные поля событий безопасности	
		Поля событий безопасности с pgAudit
24	Character count of the error position therein	Номер символа в запросе пользователя
25	Location of the error in the PostgreSQL source code	Расположение ошибки в исходном коде

6.4.3.1 Установка расширения pgAudit

Для установки расширения pgAudit необходимо:

1) Открыть конфигурационный файл:

— в ОС Windows:

```
C:\Program Files\GIS\Jatoba\<ver>\data\postgresql.conf
```

— в GNU Linux:

```
nano /var/lib/jatoba/<ver>/data/postgresql.conf
```

2) Установить параметр в конфигурационном файле в postgresql.conf:

```
shared_preload_libraries = 'pgaudit'
```

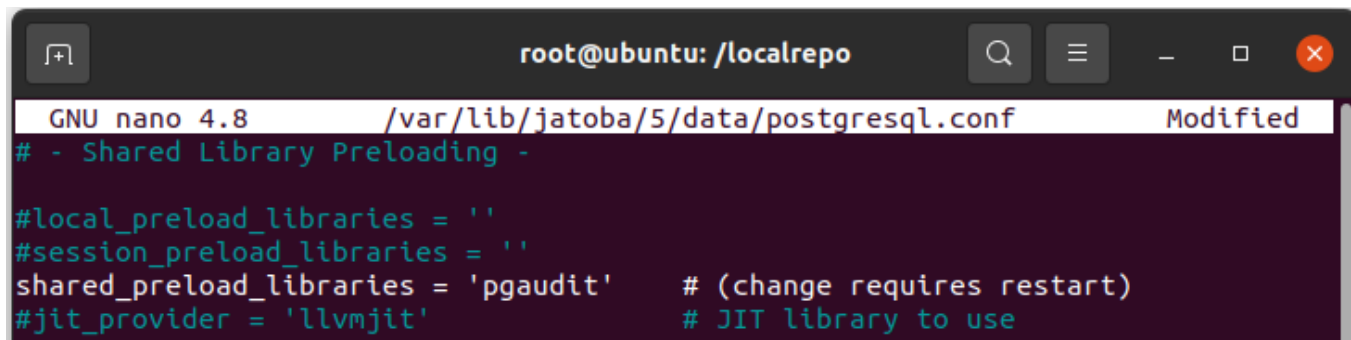


Рисунок 6.43 – Параметр загрузки библиотеки компонента «pgAudit»

3) Перезапустить СУБД «Jatoba»:

— в ОС Windows:

```
net stop JatobaServer
net start JatobaServer
```

— в GNU Linux:

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

```
systemctl restart jatoba-<ver>
```

4) Войти в СУБД от имени и с правами пользователя «SUPERUSER», выполнить SQL-команду:

```
CREATE EXTENSION pgaudit;
```

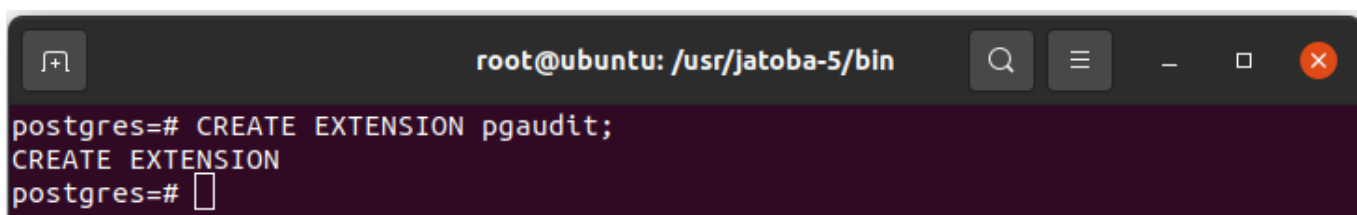


Рисунок 6.44 – Установка расширения «pgaudit»

5) Перезапустить СУБД:

— в ОС Windows:

```
net stop JatobaServer  
net start JatobaServer
```

— в GNU Linux:

```
systemctl restart jatoba-<ver>
```

После чего установку расширения можно считать оконченной.

6.4.3.2 Функциональные возможности компонента pgAudit

Для корректной работы компонента потребуется, чтобы в конфигурационном файле был установлен параметр «log_statement», как было выше описано в п. 6.4.1, 6.4.2.

pgaudit.log

Обязательно должен быть установлен параметр pgaudit.log. По умолчанию установлено значение «none». При помощи SQL-команды можно установить какие классы операторов будут регистрироваться в журнале событий.

Значения параметра могут быть следующими:

– READ – регистрируются SQL-команды SELECT, COPY в случае если источником является отношение или запрос;

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

- **WRITE** – регистрируются SQL-команды INSERT, UPDATE, DELETE, TRUNCATE, и COPY;
- **FUNCTION** – регистрируются функции CALLS и DO;
- **ROLE** – регистрируются SQL-команды относящиеся к ролям и системным привилегиям такие как, GRANT, REVOKE, CREATE/ALTER/DROP ROLE;
- **DDL** – регистрируются SQL-команды DDL не относящиеся к параметру ROLE;
- **MISC** – регистрируются прочие команды SQL-команды, такие как DISCARD, FETCH, CHECKPOINT, VACUUM, SET;
- **MISC_SET** – регистрируются SQL-команды типа SET;
- **ALL** – регистрируются все перечисленные SQL-команды.

Сравнение регистрируемых SQL-команд при стандартной регистрации событий безопасности СУБД и с применением компонента pgAudit приведены в таблице 6.19.

Таблица 6.19 – Сравнительная таблица регистрируемых SQL-команд

log_statement		pg_Audit	
Параметр	SQL-команды записываемые в журнал	Параметр	SQL-команды записываемые в журнал
ALL		ALL	
		READ	SELECT COPY TO
MOD	INSERT	WRITE	INSERT
	UPDATE		UPDATE
	DELETE		DELETE
	TRUNCATE		TRUNCATE
	COPY FROM		COPY FROM
	PREPARE		
	EXECUTE		
	EXPLAIN ANALYZE		
DDL	CREATE	DDL	CREATE
	ALTER		ALTER
	DROP		DROP
		FUNCTION	CALL
			DO
		ROLE	GRANT
			REVOKE

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

log_statement		pg_Audit	
Параметр	SQL-команды записываемые в журнал	Параметр	SQL-команды записываемые в журнал
			ALTER DEFAULT PRIVILEGES
			SET ROLE
		MISC	DISCARD
			FETCH
			CHECKPOINT
			VACUUM
			SET
		MISC_SET	SET
NONE		NONE	

pgaudit.log_catalog

Указывает, что ведение журнала сеанса должно быть включено в случае, когда все отношения в операторе находятся в pg_catalog.

Значение по умолчанию – on.

pgaudit.log_client

Указывает, будут ли сообщения журнала видны клиентскому процессу, такому как psql. Обычно этот параметр следует оставлять отключенным, но он может быть полезен для отладки или других целей.

Значение по умолчанию – off.

pgaudit.log_level

Указывает уровень детализации журнала, который будет использоваться для записей журнала.

Уровни детализации журнала FATAL и PANIC запрещены.

Этот параметр используется для регрессионного тестирования, а также может быть полезен конечным пользователям для тестирования или других целей.

Значение по умолчанию – log.

pgaudit.log_parameter

Параметр указывает, что журнал аудита должен включать параметры, которые были переданы с оператором. При наличии параметров они будут включены в CSV формат после текста оператора.

Значение по умолчанию – off.

pgaudit.log_relation

Параметр указывает должны ли отражаться в журнале регистрации событий отдельные записи для каждого отношения (TABLE, VIEW, и т.д.), указанного в операторе SELECT или DML.

Значение по умолчанию – off.

pgaudit.log_rows

Параметр указывает, что журнал аудита должен включать строки, извлеченные или затронутые оператором. Если включено, поле строк будет включено после поля параметров.

Значение по умолчанию – off.

pgaudit.log_statement

Параметр указывает, будет ли протоколирование включать текст инструкции и параметры (если включено). В зависимости от требований журнал аудита может не требовать этого, и журналы становятся менее подробными.

Значение по умолчанию – on.

pgaudit.log_statement_once

Параметр указывает, будет ли протоколирование включать текст оператора и параметры с первой записью журнала для комбинации оператора/подоператора или с каждой записью. Отключение этого параметра приведет к менее подробному журналированию, но может затруднить определение инструкции, сгенерировавшей запись в журнале, хотя пары оператор/подоператор вместе с идентификатором процесса должно быть достаточно для идентификации текста оператора, зарегистрированного с предыдущей записью.

Значение по умолчанию – off.

pgaudit.role

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

Указывает основную роль, используемую для ведения журнала аудита объектов. Можно определить несколько ролей аудита, назначив их главной роли. Это позволяет нескольким группам отвечать за различные аспекты ведения журналов аудита.

По умолчанию роли нет.

Пример № 1. Создание записи журнала аудита для всех событий

Для создания записи журнала аудита для всех событий потребуется:

- 1) От имени и с правами пользователя «Superuser» авторизоваться в СУБД:

```
psql -U postgres
```

- 2) Задать запись всех событий в журнал аудита, выполнив SQL-команду:

```
ALTER SYSTEM SET pgaudit.log = 'All';
```

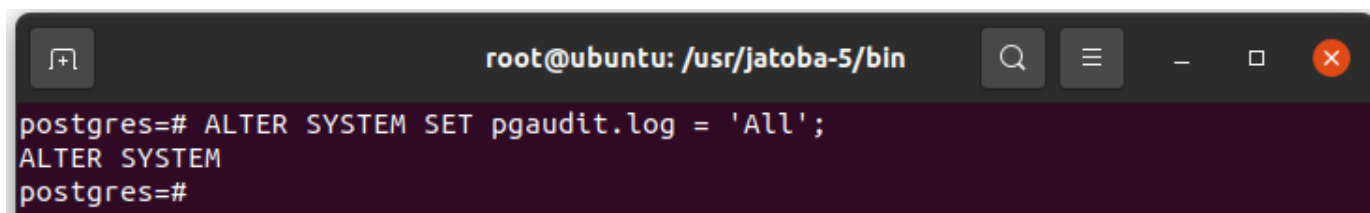


Рисунок 6.45 – Установка параметра записи всех событий в журнал аудита

- 3) Проверить заданный параметр записи всех событий:

```
SELECT name,setting FROM pg_settings WHERE name LIKE  
'pgaudit%';
```

Пример № 2. Создание записей аудита событий для определенной роли

Для создания записи журнала аудита для определенной роли потребуется:

- 1) От имени и с правами пользователя «Superuser» авторизоваться в СУБД:

```
psql -U postgres
```

- 2) Создать роль «auditor»:

```
CREATE ROLE auditor;
```

- 3) Задать запись событий «Чтение» для роли «auditor»:

```
ALTER ROLE auditor SET pgaudit.log = 'Read';
```

4) Проверить заданный параметр записи «Чтение» всех событий для роли «auditor»:

```
SELECT rolname, rolconfig from pg_roles;
```

5) Создать несколько команд в СУБД от роли «auditor»:

```
# CREATE TABLE Test (Id int NOT NULL, LastName varchar(255) NOT  
NULL, FirstName varchar(255), Age int, PRIMARY KEY (ID));  
# INSERT INTO Test (Id, LastName, FirstName, Age) VALUES  
(1, 'Testov', 'Test', 123);  
SELECT * FROM test;
```

6) Создать несколько команд в СУБД от роли «postgres»:

```
# CREATE TABLE Test1 (Id int NOT NULL, LastName varchar(255)  
NOT NULL, FirstName varchar(255), Age int, PRIMARY KEY (ID));  
# INSERT INTO Test1 (Id, LastName, FirstName, Age) VALUES  
(1, 'Testov', 'Test', 123);  
SELECT * FROM test1;
```

7) Проверить созданные записи в логе аудита:

```
nano /var/lib/jatoba/<ver>/data/log/jatoba-день_недели.log
```

Пример № 3. Создание записей аудита событий для определенной колонки в таблице

Для создания записи журнала аудита для определенной колонки в таблице потребуется:

1) От имени и с правами пользователя «Superuser» авторизоваться в СУБД:

```
psql -U postgres
```

2) Создать роль «auditor»:

```
CREATE ROLE auditor;
```

- 3) Добавить роль в таблицу «pgaudit»:

```
set pgaudit.role = 'auditor';
```

- 4) Создать таблицу «Test»:

```
CREATE TABLE Test (Id int NOT NULL, LastName varchar(255) NOT NULL, FirstName varchar(255), Age int, PRIMARY KEY (ID));
```

- 5) Добавить запись в таблицу:

```
INSERT INTO Test (Id, LastName, FirstName, Age) VALUES (1, 'Testov', 'Test', 123);
```

- 6) Задать запись событий только для выражения "UPDATE" для колонки «age» в таблице «Test» для пользователя «auditor»:

```
GRANT update (age) ON Test TO auditor;
```

- 7) Выполнить команды:

```
# UPDATE Test SET Age = 1234 WHERE Age = 123;  
# UPDATE Test SET LastName = 'Booba' WHERE LastName = 'Testov';
```

- 8) Проверить созданные записи в логе аудита:

```
nano /var/lib/jatoba/4/data/log/jatoba-день_недели.log
```

Пример № 4. Создание записей аудита событий для определенной базы

- 1) От имени и с правами пользователя «Superuser» авторизоваться в СУБД:

```
psql -U postgres
```

- 2) Создать новую БД:

```
CREATE DATABASE test1;
```

- 3) Задать запись всех событий в журнал аудита для базы данных «test1»:

```
ALTER DATABASE test1 SET pgaudit.log = 'All';
```

4) Выполнить команды:

```
# CREATE TABLE Test (Id int NOT NULL, LastName varchar(255) NOT  
NULL, FirstName varchar(255), Age int, PRIMARY KEY (ID));  
# INSERT INTO Test (Id, LastName, FirstName, Age) VALUES  
(1, 'Testov', 'Test', 123);  
SELECT * FROM test;
```

5) Подключиться к созданной БД:

```
\c test1
```

6) Создать несколько команд в СУБД:

```
# CREATE TABLE Test1 (Id int NOT NULL, LastName varchar(255)  
NOT NULL, FirstName varchar(255), Age int, PRIMARY KEY (ID));  
# INSERT INTO Test1 (Id, LastName, FirstName, Age) VALUES  
(1, 'Testov', 'Test', 123);  
SELECT * FROM test1;
```

7) Проверить созданные записи в логе аудита:

```
nano /var/lib/jatoba/<ver>/data/log/jatoba-день_недели.log
```

6.4.4. Компонент «pgauditlogtofile». Хранение событий безопасности в отдельном хранилище



Компонент не поставляется с сертифицированной версией изделия

Компонент «pgauditlogtofile» служит дополнением к компоненту «pgAudit» и предназначен для хранения событий безопасности в отдельном хранилище, в частности событий подключения и отключения к СУБД. Версия компонента – 1.5.12.

6.4.4.1 Установка расширения pgauditlogtofile

Установка пакета компонента описана в документе 643.72410666.00067-07 97 01 «Руководство по установке». Компонент выполнен в виде расширения СУБД.

Для установки расширения «pgauditlogtofile» необходимо:

1) Установить параметр загрузки библиотеки компонента в разделе «shared_preload_libraries» в конфигурационном файле postgresql.conf:

```
shared_preload_libraries = 'pgaudit, auditlogtofile'
```

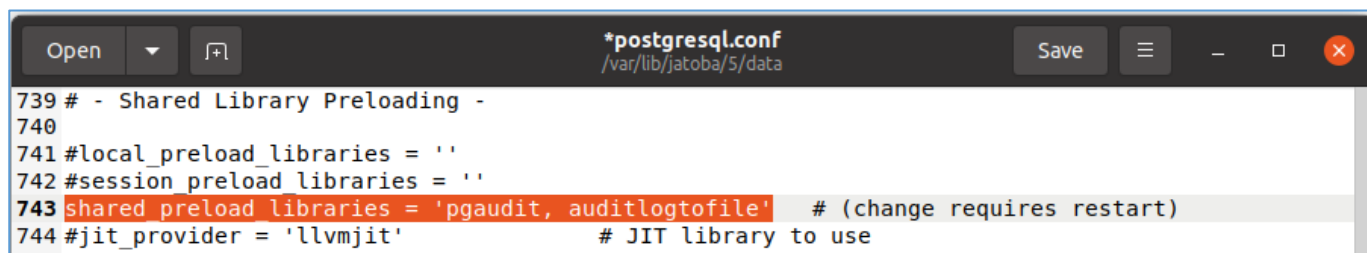


Рисунок 6.46 – Строка для загрузки библиотеки расширения «pgauditlogtofile»

Библиотека компонента «pgauditlogtofile» должна загружаться после библиотеки компонента «pgaudit».

2) Применить установленные параметры. Установленные параметры возможно применить:

- перезагрузкой службы СУБД, используя команду в терминале ОС:

```
# systemctl restart jatoba-<ver>
```

- SQL-командой в СУБД:

```
SELECT pg_reload_conf();
```

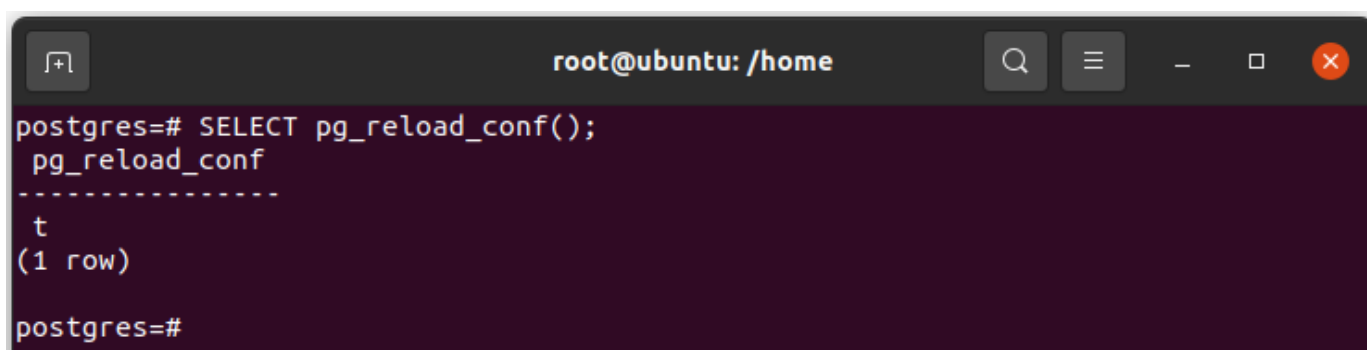


Рисунок 6.47 – SQL-команда применения изменений в конфигурационных файлах

3) Установить расширение SQL-командой:

```
CREATE EXTENSION pgauditlogtofile;
```



```

root@ubuntu: /home
postgres=# CREATE EXTENSION pgauditlogtofile;
CREATE EXTENSION
postgres=#
  
```

Рисунок 6.48 – SQL-команда установки расширения «pgauditlogtofile»

6.4.4.2 Функциональные возможности компонента pgauditlogtofile

Конфигурирование компонента проводится установкой параметров в конфигурационном файле postgresql.conf.

```

Open  *postgresql.conf  Save
/var/lib/jatoba/5/data
826 # JATOBA LOGGING PARAMETERS
827 #-----
828 log_destination = 'stderr'
829 logging_collector = on
830 log_directory = 'log'
831 log_filename = 'jatoba-%a.log'
832 log_rotation_age = 1d
833 log_rotation_size = 0
834 log_truncate_on_rotation = on
835 log_line_prefix = '%m [%p] '
836
837 pgaudit.log_directory = 'log'
838 pgaudit.log_filename = 'audit-%Y%m%d_%H%M.log'
839 pgaudit.log_rotation_age = 0
840 pgaudit.log_connections = true
841 pgaudit.log_disconnections = true
  
```

Рисунок 6.49 – Параметры компонента «pgauditlogtofile»

pgaudit.log_directory

Параметр «pgaudit.log_directory» определяет директорию хранения журнала аудита СУБД. Это ключевой параметр, который перенаправит события безопасности в отдельное хранилище.

```
pgaudit.log_directory = 'log'
```

По умолчанию установлено значение – 'log'.

В случае, если оставить параметр по умолчанию, то события безопасности отделяться не будут. Для отдельного хранения событий безопасности необходимо указать требуемый каталог. Если указываемый каталог не создан, то расширение его создаст самостоятельно.

Например

Заданная директория для хранения событий безопасности /audit_log. Для корректной работы компонента требуется установить права на доступ к директории пользователю postgres командами в терминале ОС:

```
# chown postgres: /audit_log
# chmod 700 /audit_log
# ls -ld /audit_log/
```

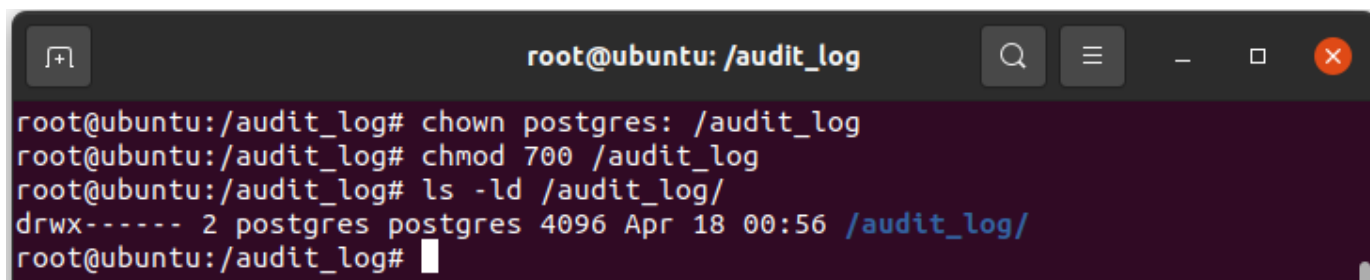


Рисунок 6.50 – Установка прав на директорию

pgaudit.log_filename

Параметр определяет имя файла, в который будет записан аудит. При записи в существующий файл будут добавлены новые записи.

```
pgaudit.log_filename = 'audit-%Y%m%d_%H%M.log'
```

По умолчанию установлено значение: 'audit-%Y%m%d_%H%M.log'

pgaudit.log_rotation_age

Параметр «pgaudit.log_rotation_age» определяет период ротации файла.

По умолчанию установлено значение – 0.

Данное значение отключает ротацию. Новые значения устанавливаются, как целые, положительные числа в минутах.

pgaudit.log_connections

Параметр определяет перехват событий безопасности соединения с СУБД.

```
pgaudit.log_connections = true
```

По умолчанию значение: false.

pgaudit.log_disconnections

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

Параметр определяет перехват событий безопасности разъединения с СУБД.

```
pgaudit.log_disconnections = true
```

По умолчанию значение: false.

6.4.5. Маскирование паролей

Функциональной возможностью СУБД является маскирование паролей в журнале аудита. Все SQL-команды, используемые в работе с СУБД, вводящие или изменяющие пароли, подвергаются процедуре маскирования паролей в журнале аудита.

В конфигурационном файле `/var/lib/jatoba/<ver>/data/postgresql.conf` обязательно должен быть установлен:

- параметр регистрации событий СУБД «log_statement», определяющий какие SQL-команды будут регистрироваться (см. таблицу 6.17);
- параметр регистрации событий СУБД «log_duration», определяющий будет ли регистрироваться длительность каждого выполненного SQL-запроса (см. таблицу 6.17);
- параметр включения/отключения маскирования «log_mask_password» который может принимать следующие значения: 1/0, on/off, true/false, yes/no.

```
log_statement='all'
log_mask_password=1
```



Рисунок 6.51 – Параметры postgresql.conf обязательные для маскирования паролей

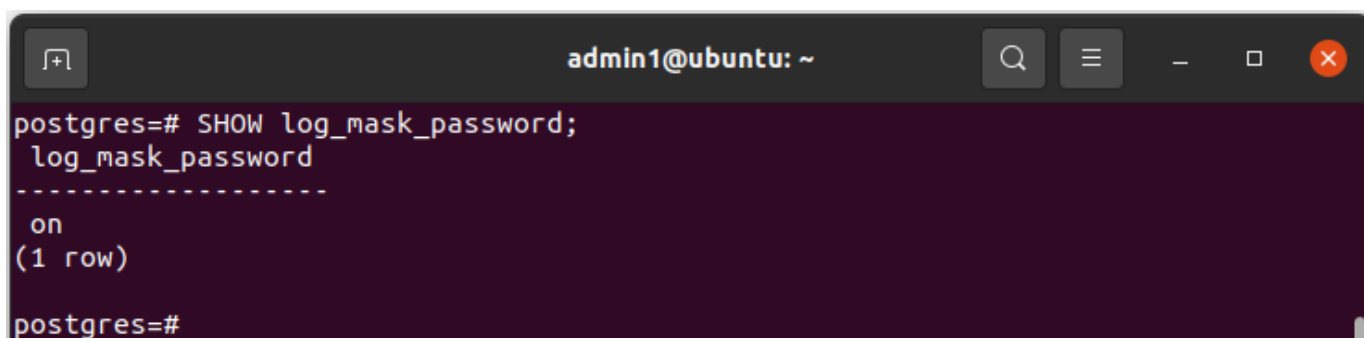
Применение установленных параметров в конфигурационных файлах выполняется от имени и справами привилегированного пользователя СУБД SQL-командой:

```
SELECT pg_reload_conf ();
```

Вторым способом применения установленных параметров является перезагрузка службы СУБД.

Установленный режим маскирования паролей выводится SQL-командой:

```
SHOW log_mask_password;
```



```
admin1@ubuntu: ~  
postgres=# SHOW log_mask_password;  
log_mask_password  
-----  
on  
(1 row)  
postgres=#
```

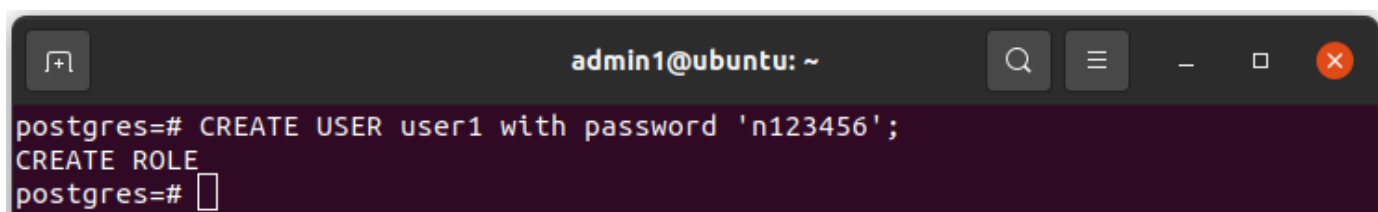
Рисунок 6.52 – Вывод режим маскирования паролей

Маскирование паролей выполняется при создании пользователей.

Например

Создается пользователь СУБД с паролем SQL-командой:

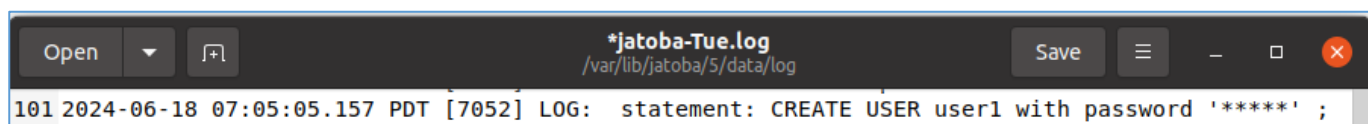
```
CREATE USER user1 with password 'n123456';
```



```
admin1@ubuntu: ~  
postgres=# CREATE USER user1 with password 'n123456';  
CREATE ROLE  
postgres=#
```

Рисунок 6.53 – SQL-команда создания пользователя

В журнале аудита СУБД установленный пароль будет маскирован.



```
*jatoba-Tue.log  
/var/lib/jatoba/5/data/log  
101 2024-06-18 07:05:05.157 PDT [7052] LOG:  statement: CREATE USER user1 with password '*****' ;
```

Рисунок 6.54 – Журнал аудита СУБД с записью создания пользователя

Аналогично маскируются вводимые значения хэшированных паролей в формате SHA256 и MD5.

Например

Установим пароль для пользователя СУБД введя хэшированные пароли SQL-запросом:

```
# ALTER ROLE user1 with password
'sha256#kvosmkwgokrpokewrpokerpokporkorgk';
# ALTER ROLE user1 with password
'md5kvosmkwgokrpokewrpokerpokporkorgk';
```

```
admin1@ubuntu: ~
postgres=# ALTER ROLE user1 with password 'sha256#kvosmkwgokrpokewrpokerpokporkorgk';
ALTER ROLE
postgres=# ALTER ROLE user1 with password 'md5kvosmkwgokrpokewrpokerpokporkorgk';
ALTER ROLE
postgres=#
```

Рисунок 6.55 – Установка хэшированного пароля пользователя

В журнале аудита СУБД установленный пароль в хэшированном формате будет маскирован.

```
*jatoba-Tue.log
/var/lib/jatoba/5/data/log
139 2024-06-18 22:41:52.874 PDT [7052] LOG:  License validation procedure has finished
140 2024-06-18 22:41:52.874 PDT [7052] LOG:  statement: ALTER ROLE user1 with password '*****' ;
141 2024-06-18 22:42:04.276 PDT [7052] LOG:  statement: ALTER ROLE user1 with password '*****' ;
```

Рисунок 6.56 – Отображение хэшированного пароля пользователя

При использовании компонента «ja_seceventlog» и сохранении событий безопасности в формате JSON, маскирование паролей будет выполняться.

```
*jsonlog-Wed.json.json
/var/lib/jatoba/5/data/log
143 {"timestamp": "2024-06-19 06:15:27.247 PDT", "user": "postgres", "dbname": "postgres", "pid": 13319, "remote_host": "[local]", "session_id": "6672c0d2.3407", "line_num": 11, "ps": "ALTER ROLE", "session_start": "2024-06-19 04:28:18 PDT", "vxid": "3/19", "txid": 744, "error_severity": "LOG", "message": "", "application_name": "psql", "backend_type": "client backend", "query_id": 0, "secEventId": 103116100, "secEventType": "УУЗП", "secEventName": "УУЗП. - 3", "secEventSubject": "postgres", "secEventLevel": "Низкий", "secEventHostName": "ubuntu", "secEventCom 11, "Substatement_id": 1, "Class": "ROLE", "Command": "ALTER ROLE", "Statement": "ALTER ROLE user1 with password <*****>", "Parameter": [{"<none>"}], "secEventResultOperation": "success", "secEventAdminRights": "yes", "secEventIp": "[local]" ["postgres", "pg_database_owner"]}]
144 {"timestamp": "2024-06-19 06:24:18.949 PDT", "user": "postgres", "dbname": "postgres", "pid":
```

Рисунок 6.57 – Маскирование пароля при использовании компонента «ja_seceventlog»

7. РЕЗЕРВНОЕ КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ БАЗ ДАННЫХ

7.1. Выгрузка кластера баз данных СУБД «Jatoba» в формате скрипта

Выгрузка кластера всех баз данных СУБД «Jatoba» в формате скрипта осуществляется с помощью следующей команды:

```
pg_dumpall -f <каталог с базы данных>all.dump
```

Более подробную информацию о параметрах `pg_dumpall` можно узнать с помощью команды:

```
pg_dumpall --help
```

7.2. Выгрузка определенной базы данных СУБД «Jatoba» в формате скрипта в файл

Выгрузка определенной базы данных СУБД «Jatoba» в формате скрипта в файл осуществляется при помощи команды:

```
pg_dump -Fc <наименование базы данных> -f <каталог с базы данных><наименование базы данных>.dump
```

Более подробную информацию о параметритах `pg_dump` можно узнать с помощью команды:

```
pg_dump --help
```

7.3. Восстановление базы данных СУБД «Jatoba» из файла архива

Восстановление базы данных СУБД «Jatoba» из файла архива, созданного командой `pg_dump`, осуществляется с помощью команды:

```
pg_restore -d <наименование базы данных> <каталог с базы данных><наименование базы данных>.dump
```

Более подробную информацию о параметритах `pg_restore` можно узнать с помощью команды:

```
pg_restore --help
```

7.4. Создание резервной копии файлов СУБД «Jatoba»

Создание резервной копии файлов СУБД «Jatoba» осуществляется при помощи команды:

```
pg_basebackup -D <каталог, где будет храниться копия файлов из каталога «data» СУБД «Jatoba»>
```

Более подробную информацию о параметрах `pg_basebackup` можно узнать с помощью команды:

```
pg_basebackup --help
```

8. НАСТРОЙКА ОТКАЗОУСТОЙЧИВОГО КЛАСТЕРА СУБД «ЯТОВА»

8.1. Настройка отказоустойчивого кластера СУБД «Jatoba» на ОС Windows Server

До настройки отказоустойчивого кластера СУБД «Jatoba» на ОС Windows Server необходимо, чтобы были выполнены следующие условия/требования:

1) Два сервера СУБД (физических или виртуальных машин) минимальные требования:

- 1xCPU 1,4 ГГц;
- RAM 6 Гб;
- HDD 50 Гб;
- 2xLAN 1 Гигабит;
- ОС Windows Server 2016.

2) Серверы СУБД должны быть в домене.

Настройка производится от имени пользователя, обладающего правами локального администратора и администратора домена, необходимо создавать записи в DNS и ActiveDirectory.

3) Два диска iSCSI-3, один минимум 600 Мб (для диска кворума кластера Windows Server Failover Cluster (далее – WSFC)), второй в зависимости от предполагаемого размера БД.

4) Серверы СУБД должны быть объединены WSFC.

Для настройки отказоустойчивого кластера СУБД «Jatoba» на ОС Windows Server 2016 необходимо выполнить следующие действия:

1) На каждом сервере в переменную среды Path добавить путь:

```
C:\Program Files\GIS\Jatoba\<ver>\bin
```

2) На первом сервере СУБД:

– если отсутствует диск Е то, через консоль диспетчера отказоустойчивости кластера, необходимо подключить диск Е (см. рисунок 8.1) к первому серверу;

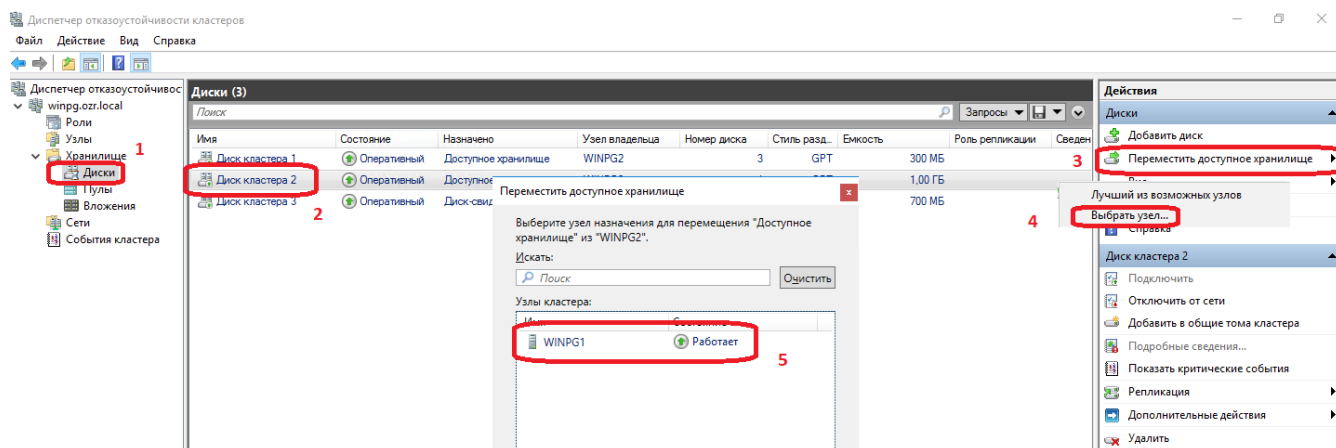


Рисунок 8.1 – Консоль диспетчера отказоустойчивости кластера. Подключение диска Е

- создать на диске Е каталог pgdata;
- запустить установку СУБД «Jatoba» из дистрибутива. Во время установки указать путь к БД е:\pgbase (см. рисунок 8.2), остальные параметры необходимо выбрать стандартно (пароль для пользователя postgres –должен быть доменный, локализация, порт и т.д.);

Data Directory

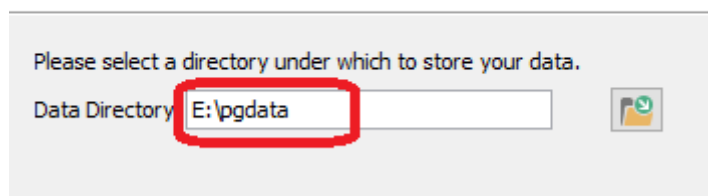


Рисунок 8.2 – Путь к БД

- остановить службу JatobaServer. Удалить содержимое каталога е:\pgdata.
- 3) Прodelать данные шаги на втором сервере, пароль и порт указать точно такие, как на первом сервере, только без остановки службы и удаления содержимого каталога е:\pgdata.
 - 4) Выдать необходимые доступы в файле е:\pgbase\ph_hba.conf.
 - 5) В файле е:\pgbase\ppstgresql.conf изменить параметр:

```
listen_addresses = '*'
```

- 6) Перезапустить службу JatobaServer.

7) Добавить роль универсальной службы. Эта роль позволяет автоматически переключать БД на другой сервер, в случае отказа первого.

8) Запустить диспетчер отказоустойчивого кластера. Добавить необходимую роль, как показано на рисунке 8.3.

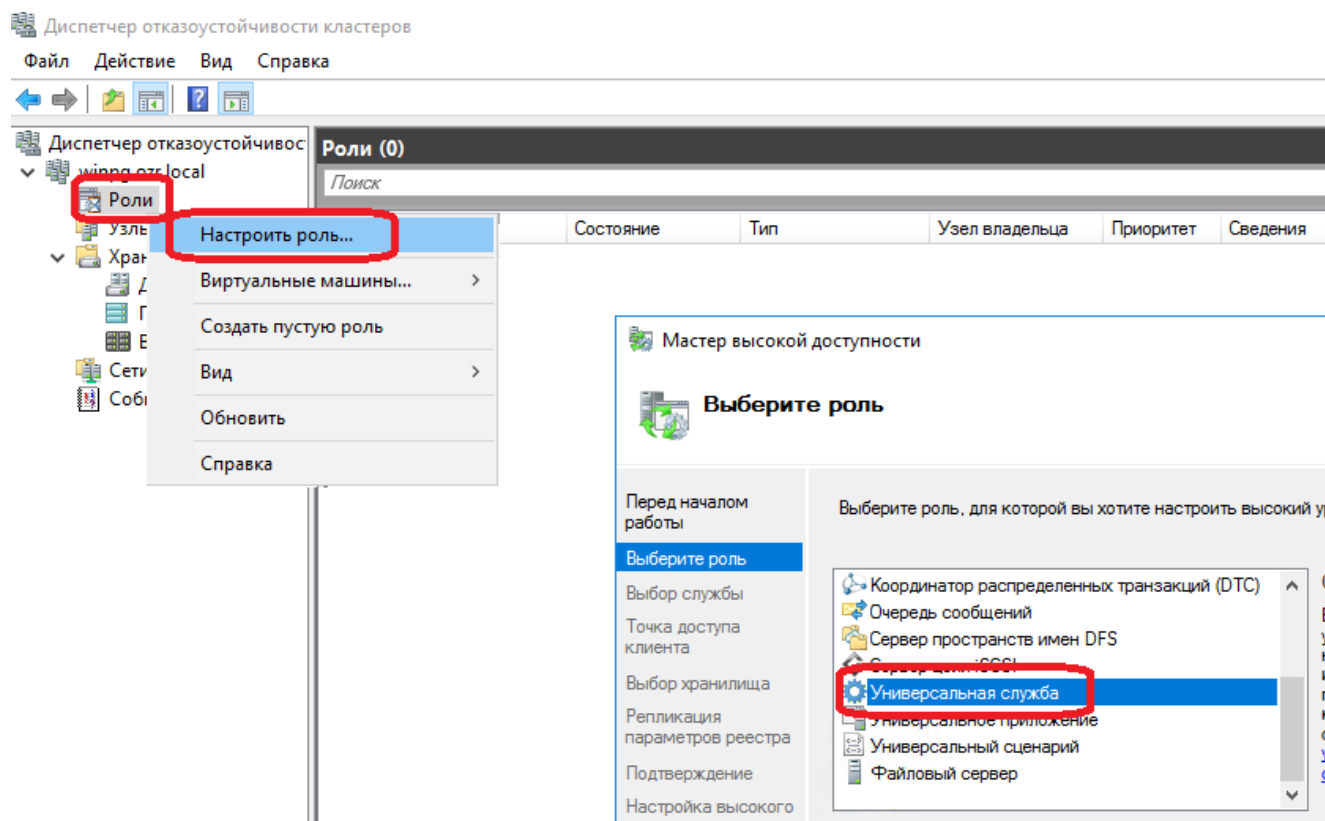


Рисунок 8.3– Добавление роли

9) Нажать на кнопку далее и выбрать службу JatobaServer, как показано на рисунке 8.4;

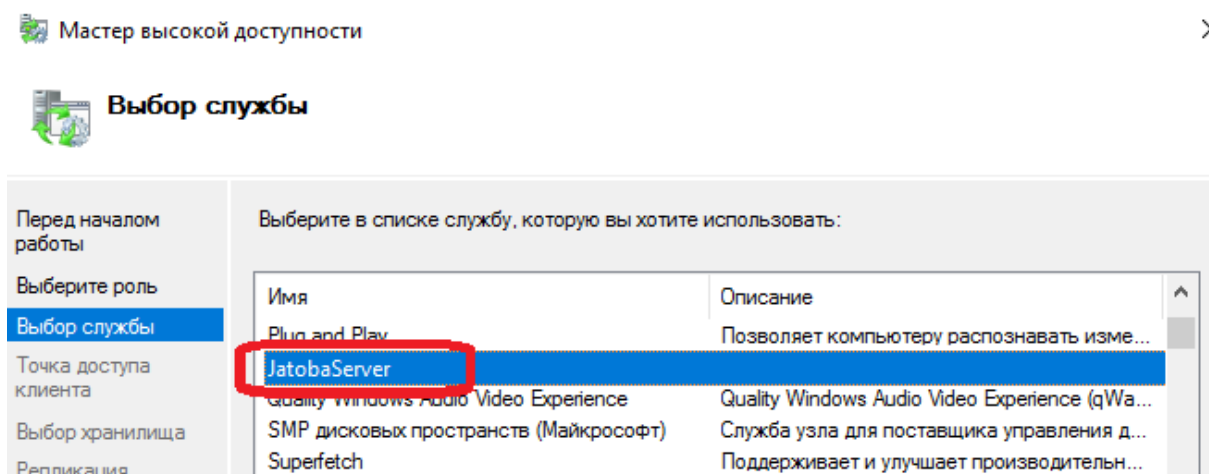


Рисунок 8.4 – Выбор службы JatobaServer

10) Нажать кнопку «Далее» и затем выбрать имя и IP-адреса точки подключения к СУБД «Jatoba» (см. пример на рисунке 8.5).

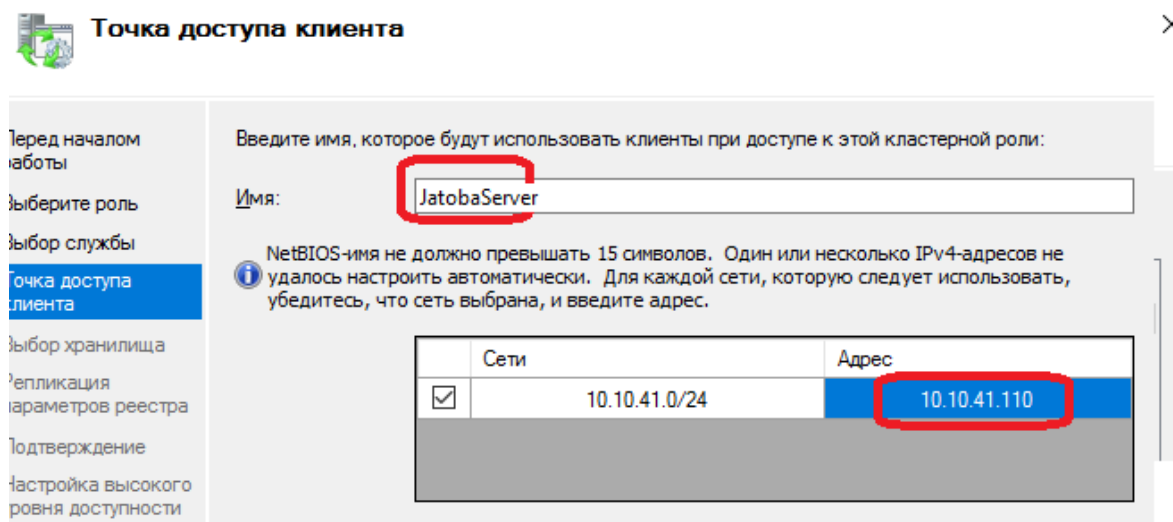


Рисунок 8.5 – Выбор имя и IP-адреса точки подключения к Jatoba

11) Нажать «Далее» и выбрать хранилище, на рисунке 8.6 – диск E.

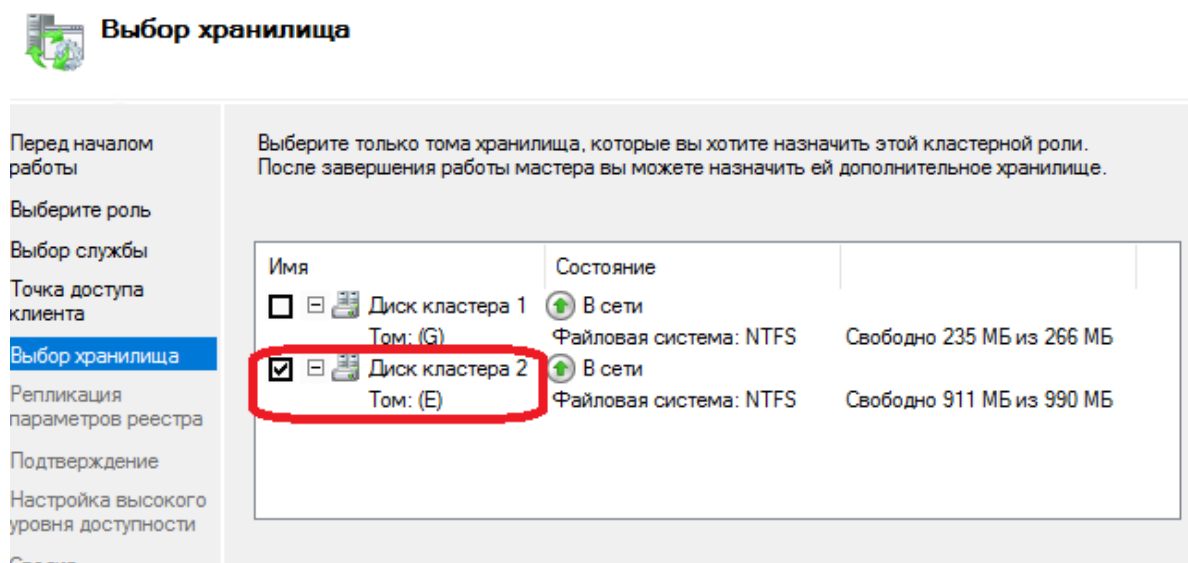


Рисунок 8.6 – Выбор хранилища

12) Нажать «Далее» и затем кнопку «Готово». На этом роль отказоустойчивости службы PG настроена.

13) Добавить контроль доступности подключения к PG.

14) Создать локального администратора на каждом сервере (в примере это пользователь rna).

15) На рабочем столе этого пользователя создать файл скрипта Powershell pg_check.ps1 со следующим содержимым:

```
$pg_status = "C:\tmp\pg_status.txt"
```

```
function pg_check {  
    $stream = [System.IO.StreamWriter] $pg_status  
  
    pg_isready -p 8080 -h 127.0.0.1 -U postgres  
    $? | % { $stream.WriteLine($_) }  
    $stream.close()  
    # Таймаут ожидания ответа от запроса pg_isready  
    Start-Sleep -Seconds 3  
}  
  
function loop_check {  
    foreach($i in 1..3) {  
        if ((Get-Content $pg_status) -eq $False) {  
            pg_check  
        }  
        # Таймаут повтора проверки запроса. При увеличении необходимо  
        # корректировать частоты выполнения скрипта в планировщике.  
        Start-Sleep -Seconds 10  
    }  
    # Перезагружаем компьютер после трех неудачных проверок pg_isready  
    Restart-Computer -force  
}  
  
pg_check  
  
if (((Get-Service -Name JatobaServer).Status -eq "Running") -and ((Get-Content  
$pg_status) -eq $False)) {  
    loop_check  
}
```

Этот скрипт проверяет доступность подключения к локальному PG и если после трех попыток не может подключиться, то перезапускает локальный Windows сервер, в результате

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

чего физические подключения к БД пойдут через второй сервер (сработает кластерная роль). Количество попыток подключения, таймаут ожидания повтора можно изменять, при увеличении таймаута необходимо увеличить время повтора выполнения скрипта в планировщике заданий.

- 16) Создать задачу в планировщике.
- 17) В задании используется именно пользователь «postgres».
- 18) Добавить триггер, как показано на рисунке 8.7.

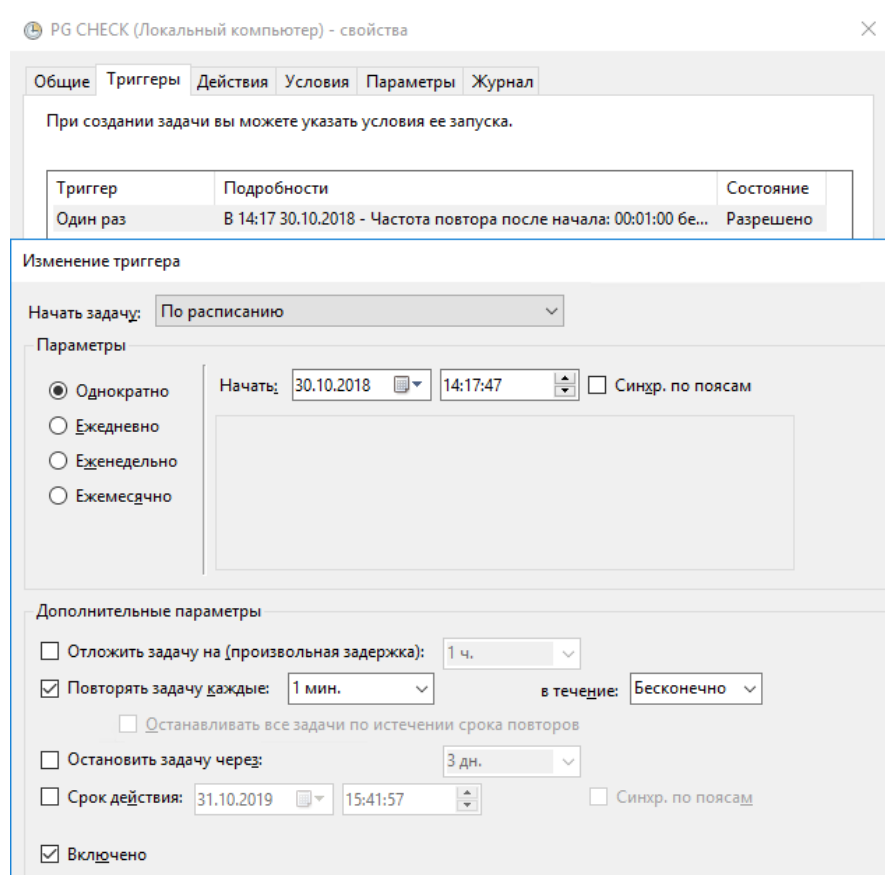


Рисунок 8.7 – Добавление триггера

- 19) Указать путь к PowerShell на локальном компьютере:

C:\Windows\System32\WindowsPowerShell\v1.1\powershell.exe

и файл скрипта как показано на рисунке 8.8:

-File "C:\tmp\pg_check.ps1",

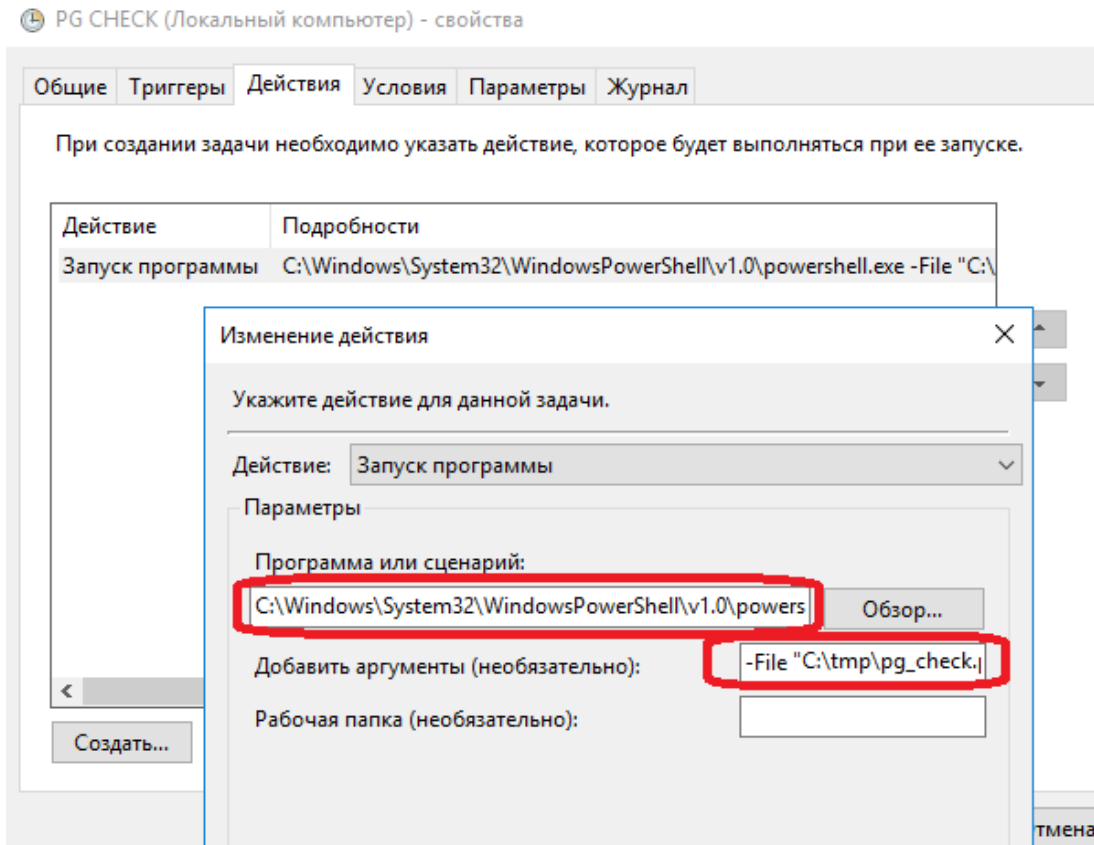


Рисунок 8.8 – Путь к PowerShell. Путь к файлу скрипта

20) Указать условия запуска, как показано на рисунке 8.9.

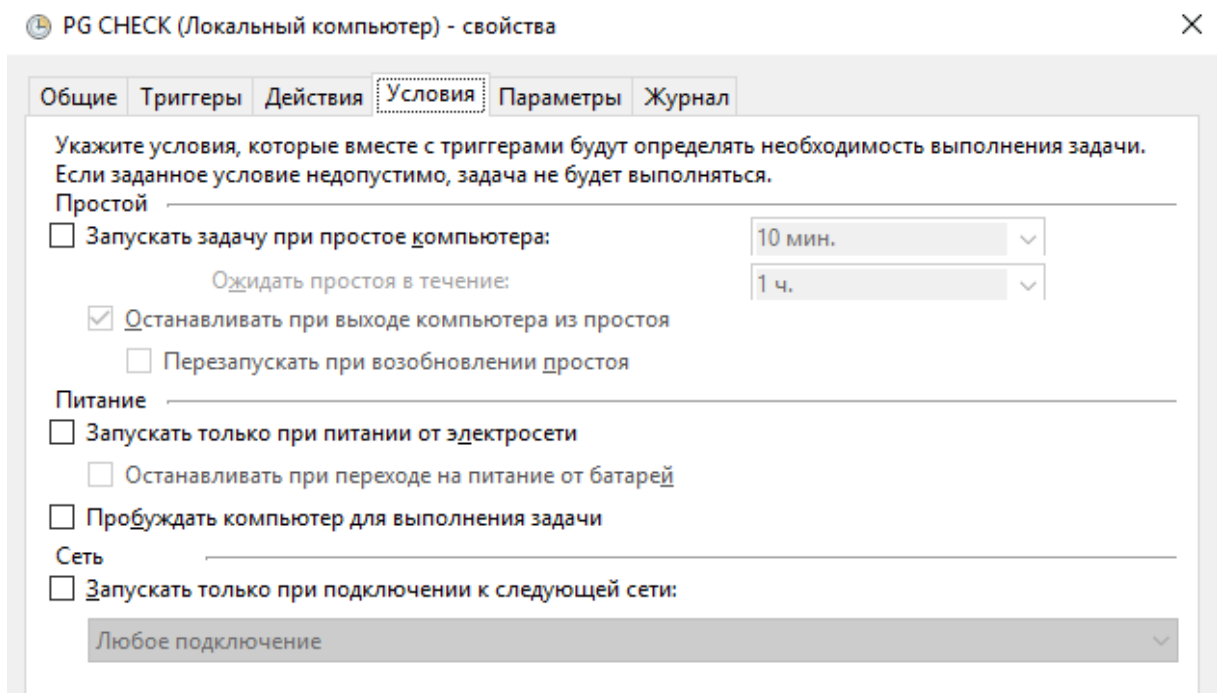


Рисунок 8.9 – Условия запуска

21) Затем параметры, как показано на рисунке 8.10.

PG CHECK (Локальный компьютер) - свойства

Общие Трiggerы Действия Условия **Параметры** Журнал

Укажите дополнительные параметры выполнения задачи.

☒ Выполнять задачу по требованию

☒ Немедленно запускать задачу, если пропущен плановый запуск

☒ При сбое выполнения перезапускать через: 1 мин.

Количество попыток перезапуска: 3

☒ Останавливать задачу, выполняемую дольше: 3 дн.

☒ Принудительная остановка задачи, если она не прекращается по запросу

☐ Если повтор задачи не запланирован, удалять через: 30 дн.

Если задача уже выполняется, то применять правило:

Не запускать новый экземпляр

Рисунок 8.10 – Параметры

8.2. Настройка отказоустойчивого кластера СУБД «Jatoba» с использованием компонента «jaDog»

Настройка отказоустойчивого кластера СУБД «Jatoba» с использованием встроенного модуля jaDog описана в документах:

- «643.72410666.00067-07 98 01-01 «Руководство по настройке. Часть 1. Управление режимом работы узлов кластера. Компонент «jaDog» (версия 1.4.2);
- «643.72410666.00067-07 98 02-01 «Руководство по настройке. Часть 1. Управление режимом работы узлов кластера. Компонент «jaDog» (версия 3.2).

8.3. Поддержка асинхронной репликации данных между несколькими БД одного и того же типа

В СУБД есть возможность асинхронной репликации данных с использованием нескольких БД одного и того же типа. Для этого применяется механизм Streaming Replication, который позволяет создавать несколько реплик одной и той же базы данных.

Настройка репликации:

- убедитесь, что все серверы, на которых будут размещены реплицированные базы данных, имеют доступ к каталогу pg_xlog/ на сервере-источнике;

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

– на сервере-источнике откройте файл `pg_hba.conf` и добавьте строку, разрешающую доступ без пароля для всех IP-адресов:

```
host replication all all 0.0.0. 0/0 md5
```

– запустите команду `pg_create_physical_replication_slot` на сервере-источнике для создания слота репликации;

Например:

```
# SELECT pg_create_physical_replication_slot('my_repl_slot');
```

– на каждом сервере, где будет размещена реплика, откройте файл `pg_hba.conf` и добавьте строку для сервера-источника, разрешающую подключение к слоту репликации, созданному на предыдущем шаге.

Например:

```
Allow replication connection from server_source
```

8.4. Поддержка задания временной задержки репликации между серверами

```
recovery_min_apply_delay (integer)
```

Параметр «`recovery_min_apply_delay`» работает путем определения минимального времени, которое должно пройти между записью операции на диск и следующей операцией записи. Это помогает обеспечить, что все операции были успешно записаны на диск, прежде чем продолжить процесс записи. Если происходит сбой системы, СУБД может использовать это время для проверки, были ли операции успешно записаны. Если время истекло, а операция все еще не была записана, СУБД попытается перезаписать ее.

По умолчанию ведомый сервер восстанавливает записи WAL передающего настолько быстро, насколько это возможно.

Задержка применяется лишь для записей WAL, представляющих фиксацию транзакций. Остальные записи проигрываются незамедлительно, так как их эффект не будет замечен до применения соответствующей записи о фиксации транзакции, благодаря правилам видимости MVCC.

Задержка добавляется, как только восстанавливаемая база данных достигает согласованного состояния, и исключается, когда ведущий сервер переключается в режим основного. После переключения ведущий сервер завершает восстановление незамедлительно.

Данный параметр предназначен для применения в конфигурациях с потоковой репликацией; однако если он задан, он будет учитываться во всех случаях, кроме восстановления после сбоя. Задержка, устанавливаемая этим параметром, влияет и на работу механизма `hot_standby_feedback`, что может привести к раздуванию базы на главном сервере; использовать данный параметр при включении этого механизма следует с осторожностью.

8.5. Развертывание отказоустойчивого кластера в приложении «Patroni» с СУБД «Jatoba»

Приложение «Patroni» можно использовать с СУБД «Jatoba» для создания высокодоступных кластеров на основе потоковой репликации., но с учетом особенностей конфигурирования.

Первой особенностью является размещение бинарных файлов и каталогов с данными. Для работы приложения «Patroni» с правильными каталогами требуется явно прописать их в конфигурации.

В разделе «`postgresql`» нужно явно задать параметры «`bin_dir`» и «`data_dir`» для путей по умолчанию:

```
bin_dir: /usr/jatoba-<ver>/bin
data_dir: /var/lib/jatoba/<ver>/data
```

Второй особенностью является инициализация СУБД.

Процедура установки СУБД описана в «Руководстве по установке. 643.72410666.00067-07 97 01». Доступна установка:

- вручную из локального репозитория;
- с помощью инсталлятора.

Инсталлятор выполнит все необходимые процедуры и СУБД будет готова для дальнейшего использования.

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

Ручная установка из локального репозитория потребует выполнения

– инициализации каталога данных СУБД:

```
./jatoba-setup initdb jatoba-<ver>
```

После этого, как будут внесены изменения в конфигурационные параметры приложения «Patroni», установлена и проинициализирована СУБД «Jatoba», можно запускать приложения «Patroni» и строить кластер.

9. ВОССТАНОВЛЕНИЕ ПОВРЕЖДЕННЫХ WAL ЗАПИСЕЙ

WAL Recovery – это функциональность ядра СУБД «Jatoba» по восстановлению поврежденных WAL записей в процессе потоковой репликации данных.

WAL – это журнал, в который попадают изменения данных до того, как они физически применяются к самой БД. Каждая запись журнала содержит информацию, достаточную для повторения изменений в случае необходимости.

При потоковой репликации с Главного сервера на Резервные передаются WAL файлы, которые при успешной проверке их контрольных сумм (CRC) применяются на Резервных узлах, за счет чего данные на Резервных узлах синхронизируются с данными на Главном сервере.

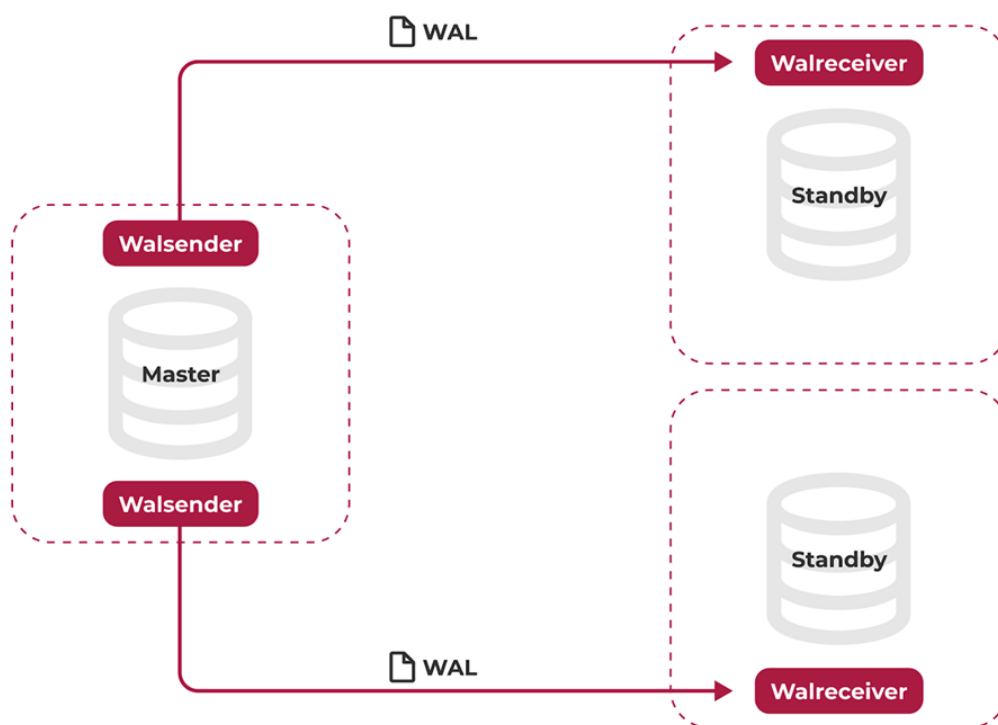


Рисунок 9.1 – Схема работы

Если на Главном узле произошла ошибка и файл журнала WAL повредился, то после передачи WAL записей на Резервные узлы, на них не совпадут контрольные суммы. В результате репликация прервется с ошибкой в логах.

Для включения восстановления поврежденных WAL записей необходимо установить параметр в файле postgresql.conf:

```
wal_sender_check_crc = on
```

Этот параметр активирует функциональность по восстановлению поврежденных WAL записей на Главном узле перед их отправкой на Резервные узлы. При успехе или неуспехе восстановления поврежденных записей WAL будет соответствующая запись в логах СУБД.

Опционально можно также установить параметр wal_sender_panic_on_crc_error:

```
wal_sender_panic_on_crc_error = on
```

При включенном параметре, сообщения об ошибке восстановления WAL записей будут записываться в лог с уровнем важности «PANIC», в противном случае уровень важности будет «FATAL».

10. ПОИСК БЛИЖАЙШИХ СОСЕДЕЙ (KNN ДЛЯ B-TREE)

Метод К-ближайших соседей (K-nearest neighbors, KNN) — это метод, используемый для решения задач классификации и регрессии. KNN основан на идее, что объекты, которые находятся рядом в пространстве признаков, вероятно относятся к одной категории.

При использовании KNN для классификации нового объекта вычисляются расстояния до всех известных объектов в наборе. Затем выбирается К объектов с наименьшими расстояниями (ближайшие соседи).

Параметр К определяет количество ближайших соседей, которые учитываются при классификации. Выбор оптимального значения К является важным этапом в настройке алгоритма KNN. Слишком малое значение К может привести к нестабильности предсказаний, в то время как слишком большое значение К может снизить точность модели, поскольку она будет учитывать слишком много неинформативных объектов.

В случае использования KNN для классификации объект присваивается тому классу, который является наиболее распространённым среди k соседей данного элемента, классы которых уже известны. В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.

Области применения метода KNN:

- предварительная обработка данных;
- механизмы рекомендаций;
- финансы: (прогнозирование фондового рынка, курсы валют, торговые фьючерсы и анализ отмывания денег);
- здравоохранение;
- распознавание образов: (идентификация шаблонов, например, при классификации текста и цифр).

Индекс B-Tree в СУБД – индекс на основе дерева. Индексы B-Tree эффективны для обычных типов данных таких как текст, числа и метки времени. Использование стандартных B-Tree индексов обычно достаточно для большинства типов данных и таблиц пользователя. Команда CREATE INDEX по умолчанию создает индекс B-Tree.

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

Структура индекса B-Tree порядка m - это дерево, удовлетворяющее требованиям:

- каждый узел имеет не более m потомков;
- каждый внутренний узел имеет как минимум $\lceil m/2 \rceil$ потомков;
- корень имеет как минимум два потомка (если он сам не лист);
- все листья находятся на одном уровне;
- не листовой узел с k потомков содержит $k-1$ ключей;
- сохраняет состояние сбалансированности и четности.

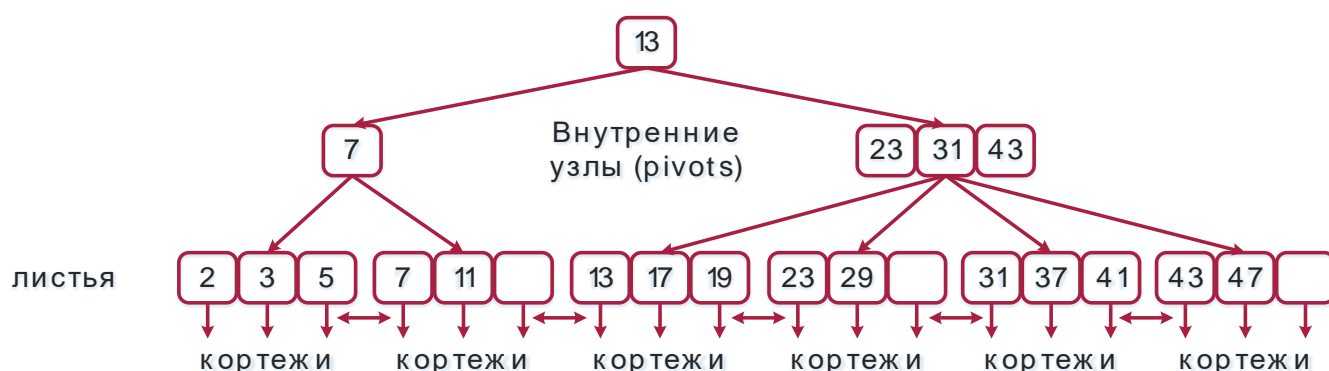


Рисунок 10.1 – Структура B-Tree

Когда выполняется поиск на основе этого индекса, он проходит вниз по дереву, чтобы найти ключ, по которому дерево построено, а затем возвращает искомые данные. Использование индекса гораздо быстрее, чем последовательное сканирование.

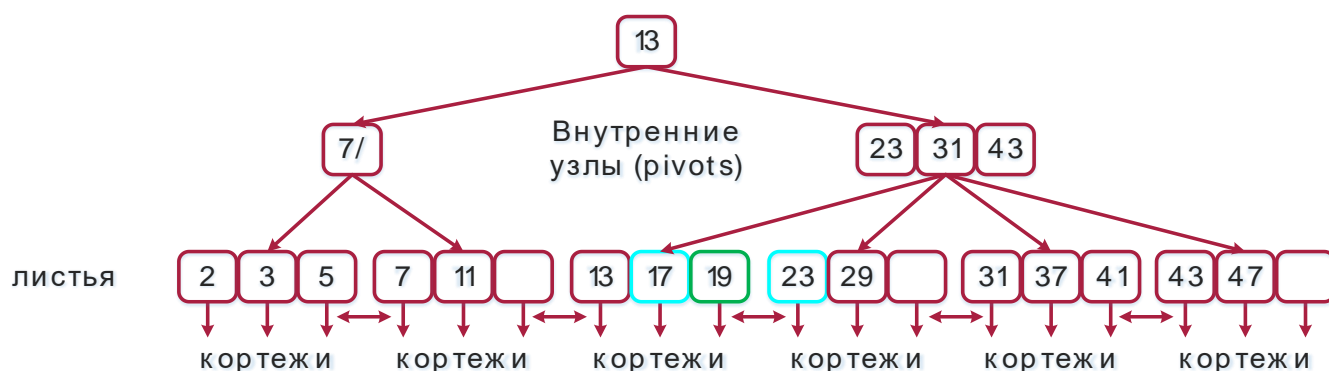


Рисунок 10.2 – K Nearest Neighbors для B-tree

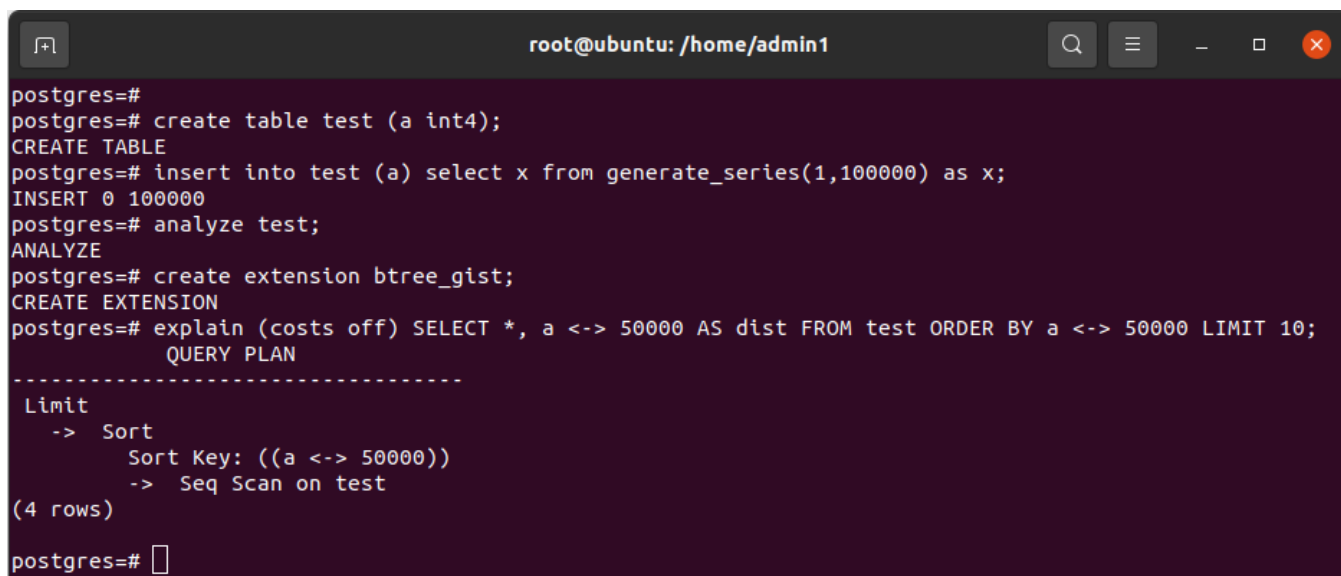
Метод KNN уже реализован для индексов типа GiST. Однако применение этих индексов не всегда оправдано для простых типов данных. Пользователь в большинстве

случаев оперирует именно простыми типами данных. В следующем примере представлена эффективность применения B-Tree индексов по сравнению GiST индексами.

Пример использования KNN с GiST

Создадим план запроса для тестовой таблицы:

```
# CREATE TABLE test (a int4);
# insert into test (a) select x from generate_series(1,100000)
as x;
# ANALYZE test;
# CREATE EXTENSION btree_gist;
# EXPLAIN (costs off) SELECT *, a <-> 50000 AS dist FROM test
ORDER BY a <-> 50000 LIMIT 10;
```

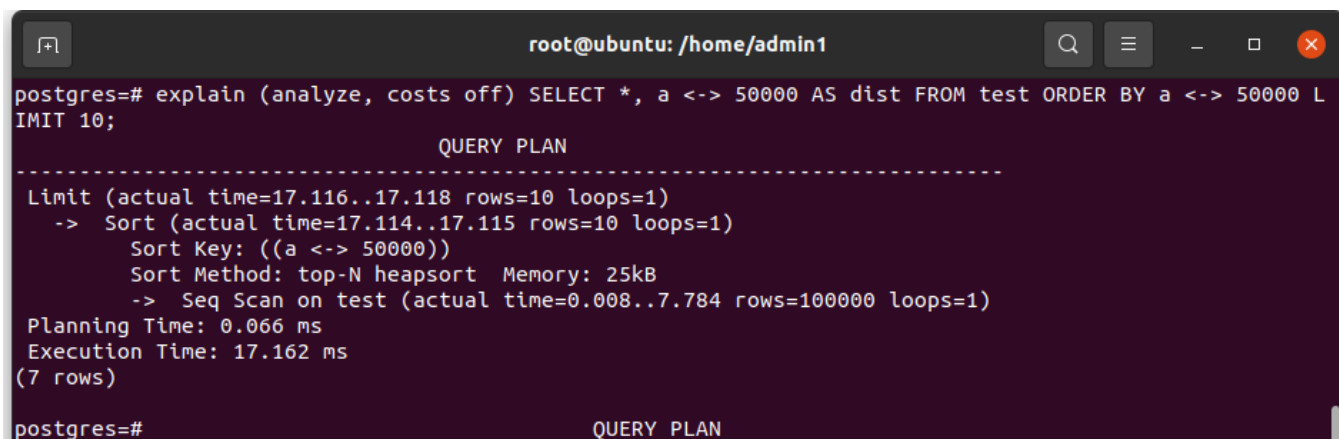


```
root@ubuntu: /home/admin1
postgres=#
postgres=# create table test (a int4);
CREATE TABLE
postgres=# insert into test (a) select x from generate_series(1,100000) as x;
INSERT 0 100000
postgres=# analyze test;
ANALYZE
postgres=# create extension btree_gist;
CREATE EXTENSION
postgres=# explain (costs off) SELECT *, a <-> 50000 AS dist FROM test ORDER BY a <-> 50000 LIMIT 10;
               QUERY PLAN
-----
Limit
  -> Sort
      Sort Key: ((a <-> 50000))
      -> Seq Scan on test
(4 rows)
postgres=#
```

Рисунок 10.3 – Создание плана запроса для тестовой таблицы

Создадим план запроса:

```
# EXPLAIN (analyze, costs off) SELECT *, a <-> 50000 AS dist
FROM test ORDER BY a <-> 50000 LIMIT 10;
```



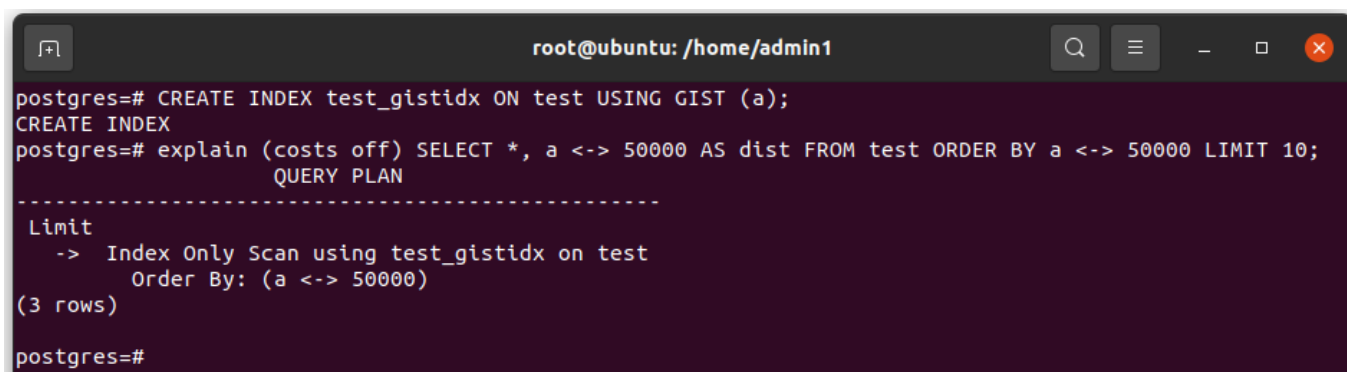
```
root@ubuntu: /home/admin1
postgres=# explain (analyze, costs off) SELECT *, a <-> 50000 AS dist FROM test ORDER BY a <-> 50000 L
IMIT 10;
               QUERY PLAN
-----
Limit (actual time=17.116..17.118 rows=10 loops=1)
  -> Sort (actual time=17.114..17.115 rows=10 loops=1)
        Sort Key: ((a <-> 50000))
        Sort Method: top-N heapsort  Memory: 25kB
  -> Seq Scan on test (actual time=0.008..7.784 rows=100000 loops=1)
Planning Time: 0.066 ms
Execution Time: 17.162 ms
(7 rows)

postgres=#
```

Рисунок 10.4 – Создание плана запроса

Создадим индекс для тестовой таблицы и план запроса:

```
# CREATE INDEX test_gistidx ON test USING GIST (a);
# explain (costs off) SELECT *, a <-> 50000 AS dist FROM test
ORDER BY a <-> 50000 LIMIT 10;
```



```
root@ubuntu: /home/admin1
postgres=# CREATE INDEX test_gistidx ON test USING GIST (a);
CREATE INDEX
postgres=# explain (costs off) SELECT *, a <-> 50000 AS dist FROM test ORDER BY a <-> 50000 LIMIT 10;
               QUERY PLAN
-----
Limit
  -> Index Only Scan using test_gistidx on test
        Order By: (a <-> 50000)
(3 rows)

postgres=#
```

Рисунок 10.5 – Создание индекса GIST

Создадим план запроса:

```
# EXPLAIN (analyze, costs off) SELECT *, a <-> 50000 AS dist
FROM test ORDER BY a <-> 50000 LIMIT 10;
```



```

root@ubuntu: /home/admin1
postgres=# explain (analyze, costs off) SELECT *, a <-> 50000 AS dist FROM test ORDER BY a <-> 50000 L
IMIT 10;
               QUERY PLAN
-----
Limit (actual time=0.082..0.085 rows=10 loops=1)
  -> Index Only Scan using test_gistidx on test (actual time=0.081..0.084 rows=10 loops=1)
        Order By: (a <-> 50000)
        Heap Fetches: 0
Planning Time: 0.043 ms
Execution Time: 0.096 ms
(6 rows)
postgres=#

```

Рисунок 10.6 – Создание плана запроса

Создадим B-Tree индекс и выведем все доступные информационные схемы (information schemas) в СУБД. Вывод показывает список таблиц, колонок, индексов, триггеров, и т.д., которые определены в каждой из информационных схем.

```

# CREATE INDEX test_btreeidx ON test USING btree (a);
# \di+

```

```

root@ubuntu: /home/admin1
postgres=# CREATE INDEX test_btreeidx ON test USING btree (a);
CREATE INDEX
postgres=# \di+
               List of relations
Schema |      Name      | Type  | Owner  | Table | Persistence | Access method | Size  | Descripti
on
-----+-----+-----+-----+-----+-----+-----+-----+-----
public | test_btreeidx  | index | postgres | test  | permanent  | btree         | 2208 kB |
public | test_gistidx   | index | postgres | test  | permanent  | gist          | 4392 kB |
(2 rows)
postgres=#

```

Рисунок 10.7 – Вывод информации о таблицах

Из примера видно следующее:

- применение стратегия поиска KNN в индексах увеличивает скорость выполнения запросов на порядки;
- индекс B-Tree работает быстрее за счет своего меньшего размера по сравнению с GIST индексом.

В СУБД «Jatoba» метод KNN реализован для B-Tree индексов. В следующем разделе описаны виды SQL-запросов, которые используют метод KNN.

10.1. Форма SQL-запроса, для которых работает KNN

Применение KNN сейчас возможно для SQL-запросов вида:

```
SELECT ... FROM ... WHERE ... ORDER BY  
проиндексированное_поле_таблицы оператор константа LIMIT n
```

или

```
SELECT (проиндексированное_поле_таблицы оператор константа) AS  
x FROM ... WHERE ... ORDER BY x LIMIT n
```

или

```
SELECT (константа оператор проиндексированное_поле_таблицы) AS  
x FROM ... WHERE ... ORDER BY x LIMIT n
```

Пример такого запроса:

Выбрать 5 ближайших к заданной дате 01/01/2024 событий из таблицы всех событий:

```
SELECT date, event, ('2024-01-01'::date <-> date) AS dist FROM  
events ORDER BY dist ASC LIMIT 5;
```

В данном примере «близость» или разница между датами событий вычисляется оператором '<->', ORDER BY сортирует эту разницу в порядке возрастания, LIMIT из сортированного списка берет первые 5 элементов.

При условии, что построен индекс по полю date, СУБД с KNN должна использовать этот индекс для поиска ближайших дат (с минимальной дистанцией) к заданной.

В плане индекса мы должны увидеть узел плана Index Scan с опцией OrderBy (обход записей по индексам), обозначающую использование введенного дополнительного поля и всего механизма KNN в целом.

Без KNN аналогичные запросы будут формироваться с альтернативными планами, производительность которых ниже или даже значительно ниже чем с KNN.

10.2. Влияние доработок по KNN на расширение btree_gist

KNN изменяет работу встроенного расширения btree_gist.

В данном виде индекса заменяется использование встроенных операторов дистанции '<->' на использование аналогичных операторов в основном системном каталоге, которые вводятся в рамках KNN. Функционал самого расширения btree_gist от этого не изменяется.

Данное изменение касается следующих типов данных:

- целые:
 - int2;
 - int4;
 - int8;
- вещественные:
 - float4;
 - float8;
- специальные:
 - oid;
 - money;
- временные:
 - date;
 - time;
 - timestamp;
 - timestamptz;
 - interval.

В данном виде индекса в функциях операторов дистанции код вычисления дистанции заменяется на вызов соответствующей функции ядра. Встроенное расширение btree_gist теперь использует эти перенесенные в ядро функции.

10.3. Влияние доработок по KNN на системный каталог

KNN вносит следующие, видимые для пользователя изменения системного каталога.

Теперь системный каталог СУБД должен содержать следующие объекты:

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

10.3.1. Системная таблица pg_amop

Системная таблица pg_amop хранит информацию об операторах семейства операторов [индексных] методов доступа (вида индекса). Семейство операторов - это группа операторов и вспомогательных процедур, обеспечивающих работу вида индекса (например, btree индекс умеет искать и сортировать данные на основании операторов <, <=, =, >=, >; вместе с некоторыми другими операторами и вспомогательными функциями эти операторы образуют семейства операторов для btree: btree/datetime_ops, btree/float_ops, ...). KNN добавляет 26 операторов (операторов дистанции '<->'), которые можно посмотреть следующим запросом:

```
SELECT
pg_amop.oid, opfamily.opfname as amopfamily , ltype.typname as
amoplefttype,
rtype.typname as amoprightright, pg_amop.amopstrategy,
pg_amop.amoppurpose,
pg_operator.oprname as amopopr, pg_am.amname as amopmethod,
opsortfamily.opfname as amopsortfamily
from pg_amop
join pg_opfamily as opfamily on pg_amop.amopfamily =
opfamily.oid
join pg_type as ltype on pg_amop.amoplefttype = ltype.oid
join pg_type as rtype on pg_amop.amoprightright = rtype.oid
join pg_operator on pg_amop.amopopr = pg_operator.oid
join pg_am on pg_amop.amopmethod = pg_am.oid
join pg_opfamily as opsorthfamily on pg_amop.amopsorthfamily =
opsorthfamily.oid
where amopstrategy=6 order by 2, 3, 4;
```

```

root@ubuntu: /home/admin1
postgres=# select
pg_amop.oid, opfamily.opfname as amopfamily, ltype.typname as amoplefttype,
rtype.typname as amoprightright, pg_amop.amopstrategy, pg_amop.amoppurpose,
pg_operator.oprname as amopopr, pg_am.amname as amopmethod, opsortfamily.opfname as amopsortfamily
from pg_amop
join pg_opfamily as opfamily on pg_amop.amopfamily = opfamily.oid
join pg_type as ltype on pg_amop.amoplefttype = ltype.oid
join pg_type as rtype on pg_amop.amoprightright = rtype.oid
join pg_operator on pg_amop.amopopr = pg_operator.oid
join pg_am on pg_amop.amopmethod = pg_am.oid
join pg_opfamily as opsortfamily on pg_amop.amopsortfamily = opsortfamily.oid
where amopstrategy=6 order by 2, 3, 4;

```

oid	amopfamily	amoplefttype	amoprightright	amopstrategy	amoppurpose	amopopr	amopmethod	amopsortfamily
10139	datetime_ops	date	date	6 o		<->	btree	integer_ops
10145	datetime_ops	date	timestamp	6 o		<->	btree	interval_ops
10151	datetime_ops	date	timestamp	6 o		<->	btree	interval_ops
10163	datetime_ops	timestamp	date	6 o		<->	btree	interval_ops
10157	datetime_ops	timestamp	timestamp	6 o		<->	btree	interval_ops
10169	datetime_ops	timestamp	timestamp	6 o		<->	btree	interval_ops
10181	datetime_ops	timestamp	date	6 o		<->	btree	interval_ops
10187	datetime_ops	timestamp	timestamp	6 o		<->	btree	interval_ops
10175	datetime_ops	timestamp	timestamp	6 o		<->	btree	interval_ops
10080	float_ops	float4	float4	6 o		<->	btree	float_ops
10086	float_ops	float4	float8	6 o		<->	btree	float_ops
10098	float_ops	float8	float4	6 o		<->	btree	float_ops
10092	float_ops	float8	float8	6 o		<->	btree	float_ops
10005	integer_ops	int2	int2	6 o		<->	btree	integer_ops
10011	integer_ops	int2	int4	6 o		<->	btree	integer_ops
10017	integer_ops	int2	int8	6 o		<->	btree	integer_ops
10029	integer_ops	int4	int2	6 o		<->	btree	integer_ops
10023	integer_ops	int4	int4	6 o		<->	btree	integer_ops
10035	integer_ops	int4	int8	6 o		<->	btree	integer_ops
10047	integer_ops	int8	int2	6 o		<->	btree	integer_ops
10053	integer_ops	int8	int4	6 o		<->	btree	integer_ops
10041	integer_ops	int8	int8	6 o		<->	btree	integer_ops
10204	interval_ops	interval	interval	6 o		<->	btree	interval_ops

--More--

Рисунок 10.8 – Список операторов дистанции

10.3.2. Системная таблица pg_operator

Системная таблица pg_operator хранит информацию об операторах, которые можно использовать в выражениях SQL-запросов (например, в выражении $a < 5$, ' $<$ ' - оператор). Оператор может быть бинарным, а именно, операнд слева (тип операнда) и операнд справа (тип операнда), а также должен возвращать какое-то значение (зависит от оператора, в примере сравнивает значения целочисленных типов и возвращает значение булевого типа). KNN для возможности выполнения описанных выше видов запросов вводит 26 новых операторов дистанции для простых типов. Список таких операторов можно получить следующим запросом:

```

SELECT
o.oid, o.oprname, ns.nspname as oprnamespace, ai.rolname as
opowner,
o.oprkind, o.oprcanmerge, o.oprcanhash, ltt.typname as oprleft,
rtt.typname as oprright,
rst.typname as oprresult, co.oprname as oprcom, no.oprname as
oprnegate, p.proname as oprcode,
pr.proname as oprrest, pj.proname as oprjoin
from pg_operator as o
join pg_namespace as ns on ns.oid = o.oprnamespace

```

```

join pg_authid as ai on ai.oid = o.oprowner
join pg_type as ltt on ltt.oid = o.oprleft
join pg_type as rtt on rtt.oid = o.oprright
join pg_type as rst on rst.oid = o.oprresult
left join pg_operator as co on co.oid = o.oprcom
left join pg_operator as no on no.oid = o.oprnegate
left join pg_proc as p on p.oid = o.oprcode
left join pg_proc as pr on pr.oid = o.oprrest
left join pg_proc as pj on pj.oid = o.oprjoin
where
o.oprname = '<->' and
array[ltt.typname::text, rtt.typname::text, rst.typname::text]
<@ array['int2', 'int4', 'int8', 'oid', 'float4', 'float8',
'money', 'date', 'time', 'timestamp', 'timestampz',
'interval']
order by 8, 9, 10;

```

```

root@ubuntu: /home/admin1
postgres=# select
o.oid, o.oprname, ns.nspname as oprnamespace, ai.rolname as oprowner,
o.oprkind, o.oprcanmerge, o.oprcanhash, ltt.typname as oprleft, rtt.typname as oprright,
rst.typname as oprresult, co.oprname as oprcom, no.oprname as oprnegate, p.oprcode,
pr.oprcode as oprrest, pj.oprcode as oprjoin
from pg_operator as o
join pg_namespace as ns on ns.oid = o.oprnamespace
join pg_authid as ai on ai.oid = o.oprowner
join pg_type as ltt on ltt.oid = o.oprleft
join pg_type as rtt on rtt.oid = o.oprright
join pg_type as rst on rst.oid = o.oprresult
left join pg_operator as co on co.oid = o.oprcom
left join pg_operator as no on no.oid = o.oprnegate
left join pg_proc as p on p.oid = o.oprcode
left join pg_proc as pr on pr.oid = o.oprrest
left join pg_proc as pj on pj.oid = o.oprjoin
where
o.oprname = '<->' and
array[ltt.typname::text, rtt.typname::text, rst.typname::text] <@ array['int2', 'int4', 'int8', 'oid', 'float4', 'float8', 'money', 'date', 'time', 'timestamp', 'timestampz', 'interval']
order by 8, 9, 10;
oid | oprname | oprnamespace | oprowner | oprkind | oprcanmerge | oprcanhash | oprleft | oprright | oprresult | oprcom | oprnegate | oprcode | oprrest | oprjoin
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
9454 | <-> | pg_catalog | postgres | b | f | f | date | date | int4 | <-> | <-> | date_distance | <-> | <->
9440 | <-> | pg_catalog | postgres | b | f | f | date | timestamp | interval | <-> | <-> | date_dist_timestamp | <-> | <->
9442 | <-> | pg_catalog | postgres | b | f | f | date | timestampz | interval | <-> | <-> | date_dist_timestampz | <-> | <->
9451 | <-> | pg_catalog | postgres | b | f | f | float4 | float4 | float4 | <-> | <-> | float4dist | <-> | <->
9438 | <-> | pg_catalog | postgres | b | f | f | float4 | float8 | float8 | <-> | <-> | float48dist | <-> | <->
9439 | <-> | pg_catalog | postgres | b | f | f | float8 | float4 | float8 | <-> | <-> | float84dist | <-> | <->
9452 | <-> | pg_catalog | postgres | b | f | f | float8 | float8 | float8 | <-> | <-> | float88dist | <-> | <->
9447 | <-> | pg_catalog | postgres | b | f | f | int2 | int2 | int2 | <-> | <-> | int2dist | <-> | <->
9432 | <-> | pg_catalog | postgres | b | f | f | int2 | int4 | int4 | <-> | <-> | int24dist | <-> | <->
9434 | <-> | pg_catalog | postgres | b | f | f | int2 | int8 | int8 | <-> | <-> | int28dist | <-> | <->
9433 | <-> | pg_catalog | postgres | b | f | f | int4 | int2 | int4 | <-> | <-> | int42dist | <-> | <->
9448 | <-> | pg_catalog | postgres | b | f | f | int4 | int4 | int4 | <-> | <-> | int4dist | <-> | <->
9436 | <-> | pg_catalog | postgres | b | f | f | int4 | int8 | int8 | <-> | <-> | int48dist | <-> | <->
9435 | <-> | pg_catalog | postgres | b | f | f | int8 | int2 | int8 | <-> | <-> | int82dist | <-> | <->
9437 | <-> | pg_catalog | postgres | b | f | f | int8 | int4 | int8 | <-> | <-> | int84dist | <-> | <->
9449 | <-> | pg_catalog | postgres | b | f | f | int8 | int8 | int8 | <-> | <-> | int8dist | <-> | <->
9458 | <-> | pg_catalog | postgres | b | f | f | interval | interval | interval | <-> | <-> | interval_distance | <-> | <->
9453 | <-> | pg_catalog | postgres | b | f | f | money | money | money | <-> | <-> | cash_distance | <-> | <->
9450 | <-> | pg_catalog | postgres | b | f | f | oid | oid | oid | <-> | <-> | oiddist | <-> | <->
9455 | <-> | pg_catalog | postgres | b | f | f | time | time | interval | <-> | <-> | time_distance | <-> | <->
9441 | <-> | pg_catalog | postgres | b | f | f | timestamp | date | interval | <-> | <-> | timestamp_dist_date | <-> | <->
9456 | <-> | pg_catalog | postgres | b | f | f | timestamp | timestamp | interval | <-> | <-> | timestamp_distance | <-> | <->
9444 | <-> | pg_catalog | postgres | b | f | f | timestamp | timestampz | interval | <-> | <-> | timestamp_dist_timestampz | <-> | <->
9443 | <-> | pg_catalog | postgres | b | f | f | timestampz | date | interval | <-> | <-> | timestampz_dist_date | <-> | <->
9445 | <-> | pg_catalog | postgres | b | f | f | timestampz | timestamp | interval | <-> | <-> | timestampz_dist_timestamp | <-> | <->
9457 | <-> | pg_catalog | postgres | b | f | f | timestampz | timestampz | interval | <-> | <-> | timestampz_distance | <-> | <->
(26 rows)
postgres=#

```

Рисунок 10.9 – Вывод операторов дистанции для простых типов.

10.3.3. Системная таблица pg_proc

Системная таблица `pg_proc` хранит информацию о функциях, которые можно использовать в SQL-запросах. Под каждый оператор (специальные знаки в выражениях) также подставляется какая-либо функция, которая выполняет конечный код. KNN для возможности выполнения описанных выше видов запросов (выполнения операторов дистанции) вводит 26 новых функций, которые вычисляют дистанцию между двумя

значениями перечисленных выше типов данных, и которые можно посмотреть следующим запросом:

```
SELECT
p.oid, p.proname, p.provolatile, t.typname as proretype,
(select string_agg(t.typname, ' ') from pg_type as t join
lateral unnest(p.proargtypes) as pat on t.oid = pat) as
proargtypes,
p.prosrc
from pg_proc as p
join pg_type as t on p.proretype = t.oid
where
p.proname = any(array['int2dist', 'int4dist', 'int8dist',
'oiddist', 'float4dist', 'float8dist', 'cash_distance',
'date_distance', 'time_distance', 'timestamp_distance',
'timestamptz_distance', 'interval_distance', 'int24dist',
'int28dist', 'int42dist', 'int48dist', 'int82dist',
'int84dist', 'float48dist', 'float84dist',
'date_dist_timestamp', 'date_dist_timestamptz',
'timestamp_dist_date', 'timestamp_dist_timestamptz',
'timestamptz_dist_date', 'timestamptz_dist_timestamp'])
order by 5, 4;
```

```
postgres=# select
postgres-# p.oid, p.proname, p.provolatile, t.typname as proretype,
postgres-# (select string_agg(t.typname, ' ') from pg_type as t join lateral unnest(p.proargtypes) as pat on t.oid = pat) as proargtypes,
postgres-# p.prosrc
postgres-# from pg_proc as p
postgres-# join pg_type as t on p.proretype = t.oid
postgres-# where
postgres-# p.proname = any(array['int2dist', 'int4dist', 'int8dist', 'oiddist', 'float4dist', 'float8dist', 'cash_distance', 'date_distance', 'time_distance', 'timestamp_
distance', 'timestamptz_distance', 'interval_distance', 'int24dist', 'int28dist', 'int42dist', 'int48dist', 'int82dist', 'int84dist', 'float48dist', 'float84dist', 'date_di
st_timestamp', 'date_dist_timestamptz', 'timestamp_dist_date', 'timestamp_dist_timestamptz', 'timestamptz_dist_date', 'timestamptz_dist_timestamp'])
postgres-# order by 5, 4;
 oid | proname | provolatile | proretype | proargtypes | prosrc
-----+-----+-----+-----+-----+-----
 9413 | date_distance | i | int4 | date date | date_distance
 9426 | date_dist_timestamp | i | interval | date timestamp | date_dist_timestamp
 9428 | timestamp_dist_date | i | interval | date timestamp | timestamp_dist_date
 9427 | date_dist_timestamptz | s | interval | date timestamptz | date_dist_timestamptz
 9430 | timestamptz_dist_date | s | interval | date timestamptz | timestamptz_dist_date
 9410 | float4dist | i | float4 | float4 float4 | float4dist
 9425 | float84dist | i | float8 | float4 float8 | float84dist
 9424 | float48dist | i | float8 | float4 float8 | float48dist
 9411 | float8dist | i | float8 | float8 float8 | float8dist
 9406 | int2dist | i | int2 | int2 int2 | int2dist
 9420 | int42dist | i | int4 | int2 int4 | int42dist
 9418 | int24dist | i | int4 | int2 int4 | int24dist
 9407 | int4dist | i | int4 | int4 int4 | int4dist
 9419 | int28dist | i | int8 | int8 int2 | int28dist
 9422 | int82dist | i | int8 | int8 int2 | int82dist
 9421 | int48dist | i | int8 | int8 int4 | int48dist
 9423 | int84dist | i | int8 | int8 int4 | int84dist
 9408 | int8dist | i | int8 | int8 int8 | int8dist
 9417 | interval_distance | i | interval | interval interval | interval_distance
 9412 | cash_distance | i | money | money money | cash_distance
 9409 | oiddist | i | oid | oid oid | oiddist
 9414 | time_distance | i | interval | time time | time_distance
 9415 | timestamp_distance | i | interval | timestamp timestamp | timestamp_distance
 9429 | timestamp_dist_timestamptz | s | interval | timestamp timestamptz | timestamp_dist_timestamptz
 9431 | timestamptz_dist_timestamp | s | interval | timestamp timestamptz | timestamptz_dist_timestamp
 9416 | timestamptz_distance | i | interval | timestamptz timestamptz | timestamptz_distance
(26 rows)

postgres=#
```

Рисунок 10.10 – Вывод функций

11. ОЧИСТКА ПАМЯТИ В СУБД

Согласно нормативному документу «Требования по безопасности информации к системам управления базами данных (выписка)», утвержденному приказом ФСТЭК России от 14.04.2023 № 64, необходимо выполнять очистку памяти в СУБД.

11.1. Очистка памяти СУБД средствами ОС

В текущей реализации для очистки памяти СУБД «Jatoba» используются средства сертифицированной операционной системы. Поддерживаемые сертифицированные операционные системы приведены в таблице 1.1, а используемые в них утилиты для очистки памяти приведены в таблице 11.1.

Таблица 11.1 – Утилиты, используемые в ОС

№	ОС	Наименование утилиты в ОС
1	Альт 8 СП	sfill
2	РОСА 7.3 Кобальт для серверных систем	wipe
		shred
3	Astra Linux	fly-admin-smc
4	РЕД ОС 7.3 Муром	chattr
		sfill

11.2. Очистка памяти СУБД встроенными средствами

СУБД «Jatoba» поддерживает очистку объектов доступа путем многократной перезаписи уничтожаемых (стираемых) объектов файловой системы специальными битовыми последовательностями.

Специальное удаление объектов поддерживает в следующих командах:

- DROP TABLE (включая партиционирование и каскадное партиционирование);
- DROP TEMPORARY TABLE;
- DROP MATERIALIZED VIEW;
- DROP INDEX;
- TRUNCATE;
- DROP DATABASE;
- DROP SCHEMA;
- DROP TABLESPACE;

- VACUUM FULL;
- REINDEX;
- ALTER TABLE ADD COLUMN (со значением по умолчанию);
- ALTER TABLE ALTER COLUMN TYPE;

А также поддерживаться для больших таблиц, когда объект занимает более одного сегмента файлах данных (более 1ГБ).

Очистка памяти включается через:

- конфигурационный файл «postgresql.conf»;

Потребуется, от имени привилегированного пользователя, открыть для редактирования конфигурационный файл «postgresql.conf»:

```
nano /var/lib/jatoba/6/data/postgresql.conf
```

Раскомментировать параметр «wipe_file» и установить значение «on»:

```
wipe_file = on
```

Перезагрузить СУБД.

- SQL- команду «ALTER SYSTEM».

Потребуется, от имени привилегированного пользователя, выполнить SQL-команду:

```
ALTER SYSTEM SET wipe_file TO 'on';
```

Перечитать конфигурацию:

```
SELECT pg_reload_conf();
```

Проверить значение по умолчанию для параметра:

```
SHOW wipe_file;
```

Включение режима очистки памяти автоматически включит параметр wipe_file_mask.

По умолчанию значение: 0x55 = b 0101 0101

Допустимо дополнительно установить его в конфигурационном файле «postgresql.conf»:

```
wipe_file_mask = '0x55'
```

При выводе SQL-команды:

```
SHOW wipe_file_mask;
```

Будет показано значение «wipe_file_mask: 85». Разность значений обусловлена переводом их из восьмеричной в десятичную систему.



Значение wipe_file_mask = '0x55' не рекомендуется изменять, т.к. оно является оптимальным и устанавливает битовое значение перезаписи b 0101 0101

Выполнение очистки памяти СУБД регистрируется в журнале событий.

Текст сообщения:

```
Wiping file: %s
```

Где %s - имя зачищенного файла.

12. КОМПОНЕНТ TSVECTOR2

Компонент `tsvector2` предназначен для обеспечения полнотекстового поиска в БД и предоставляет расширенный тип данных `tsvector`.

Компонент выполнен в виде расширения.

Расширение устанавливается после установки пакета компонента, как описано в документе «Руководство по установке».

Расширение компонента устанавливается от имени и с правами привилегированного пользователя SQL-командой:

```
CREATE EXTENSION tsvector2;
```

Расширенный тип данных `tsvector` реализован для обеспечения лучшего сжатия и устранения ограничения размера оригинального типа `tsvector` на 1 МБ.

Он может использоваться как прозрачная замена оригинального `tsvector` и поддерживает все его функции, операторы и типы индексов. Функции, названия которых содержат `tsvector`, были изменены на `tsvector2`.

Компонент обладает специальными функциями:

- `to_tsvector2` (from text, json, jsonb types) - преобразование текста в `tsvector2` (см. п.п. 12.1);
- `array_to_tsvector2` - преобразования из массива в строку (12.1.5);
- `tsvector2_to_array` - преобразование из строки в массив (12.1.6);
- `tsvector2_stat` - получение статистики по лексемам (12.1.7);
- `jsonb_to_tsvector2` - преобразование `jsonb` в `tsvector2` (12.1.4);
- `json_to_tsvector2` - преобразование `json` в `tsvector2` (12.1.4);
- `tsvector2_update_trigger` – обновление тригера (12.1.8);
- `tsvector2_update_trigger_column` – обновление тригера в колонке (12.1.8).

Общие функции, которые можно безопасно использовать в обоих типах:

- `strip`;

- unnest;
- length;
- setweight;
- ts_rank;
- ts_rank_cd;
- ts_delete;
- ts_filter.

12.1. Примеры использования компонента tsvector2

12.1.1. Функция «to_tsvector2». Преобразование текста в tsvector2

Выполнить нормализацию на англ. (преобразование в лексемы) слов в предложении:

```
SELECT to_tsvector2('english', 'The quick brown fox jumps over  
the lazy dog.');
```

В этом шаге слова были нормализованы и отсортированы. Стоп-слова "the" и "over" (предлоги, артикли, окончания) были исключены.

Вывод в терминале:

```
to_tsvector -----  
'brown':3 'dog':9 'fox':4 'jump':5 'lazi':8 'quick':2  
(1 row)
```

Выполнить нормализацию на русский язык (преобразование в лексемы) слов в предложении:



Перед выполнением на ОС Windows выполнить команду в термине - chcp 1251

```
SELECT to_tsvector2('russian', 'Это пример текста для  
тестирования.');
```

В этом шаге слова были нормализованы и отсортированы. Стоп-слова "the" и "over" (предлоги, артикли, окончания) были исключены.

Вывод в терминале:

```
to_tsvector
```

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

```
-----  
'пример':2 'текст':3 'тестирован':5 'эт':1
```

Преобразования текста в tsvector2:

```
SELECT to_tsvector2('This is a simple test document.') AS  
result;
```

В этом шаге слова были нормализованы и отсортированы. Стоп-слова "the" и "over" (предлоги, артикли, окончания) были исключены.

Вывод в терминале:

```
result -----  
'document':6 'simpl':4 'test':5 (1 row)
```

12.1.2. Индексация с использованием tsvector

Создать таблицу:

```
CREATE TABLE documents ( id SERIAL PRIMARY KEY, data JSONB );
```

Вставить данные:

```
INSERT INTO documents (data) VALUES ('{"title": "PostgreSQL for  
Beginners", "content": "This is a guide to PostgreSQL."}'),  
({'"title": "Advanced PostgreSQL", "content": "Learn advanced  
features of PostgreSQL."});
```

Создать индекс на колонку, который содержит данные типа tsvector, для ускорения поиска:

```
CREATE INDEX idx_gin_documents ON documents USING GIN (data);  
CREATE INDEX
```

Выключить sec scan:

```
SET enable_seqscan=off;  
SET enable_indexscan=on;  
SET enable_indexonlyscan=on;
```

Выполнить запрос с оператором сравнения для поиска искомого значения:

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

```
SELECT * FROM documents WHERE to_tsvector2('english', data->>'title') || to_tsvector2('english', data->>'content') @@ to_tsquery('PostgreSQL & Beginners');
```

Выводятся искомые значения из колонки которые были введены для tsquery запроса.

Вывод в терминал:

```
[ RECORD 1 ]
-----
id | 1 data | {"title": "PostgreSQL for Beginners", "content": "This is a guide to PostgreSQL."}
```

Убедиться, что поиск производится по индексу:

```
EXPLAIN ANALYZE SELECT * FROM documents WHERE to_tsvector2('english', data->>'title') || to_tsvector2('english', data->>'content') @@ to_tsquery('PostgreSQL & Beginners');
```

При выполнении SQL-команды SELECT в запросе используется INDEX SCAN.

12.1.3. Поиск с использованием tsvector2

Создать таблицу:

```
CREATE TABLE documents ( id SERIAL PRIMARY KEY, title TEXT, content TEXT, tsv_content TSVECTOR2 GENERATED ALWAYS AS (to_tsvector2('english', content)) STORED );
```

Вставить данные:

```
INSERT INTO documents (title, content) VALUES ('PostgreSQL Basics', 'This document explains the basics of PostgreSQL.'), ('Advanced PostgreSQL', 'Learn advanced features of PostgreSQL for performance tuning.'), ('Full Text Search in PostgreSQL', 'This article covers full text search capabilities in PostgreSQL.');
```

Выполнить запрос для проверки полнотекстового поиска:

```
SELECT * FROM documents WHERE tsv_content @@ to_tsquery('PostgreSQL & basics');
```

Выводится документ с заголовком "PostgreSQL Basics".

Вставить данные:

```
INSERT INTO documents (title, content) VALUES  
('Learning text', 'This is test text for testing jatoba');
```

Выполнить запрос:

```
SELECT * FROM documents WHERE tsv_content @@ to_tsquery('test &  
jatoba');
```

Должен вернуть документ с заголовком:

```
"Learning text | This is test text for testing jatoba".
```

Выполнить запрос для разбора и нормализации текстового содержимого документа при помощи to_tsvector2 и произвести поиск

```
SELECT to_tsvector2('fat cats ate fat rats') @@ to_tsquery('fat  
& rat');
```

В ответе возвращается true - так как значение типа tsquery содержит искомые слова, это должны быть уже нормализованные лексемы

Ответ в терминале:

```
?column?  
-----  
t
```

Выполнить запрос для разбора и нормализации текстового содержимого документа при помощи to_tsvector2 и произвести поиск:

```
SELECT to_tsvector2('fatal error') @@ to_tsquery('fatal &  
error');
```

Вывод:

```
?column?  
-----  
t
```

12.1.4. Функции «jsonb_to_tsvector2» и «json_to_tsvector2». Преобразование json и jsonb в tsvector2

Выполнить нормализацию (преобразование в лексемы) значений в ключах jsonb:

```
SELECT to_tsvector2('{ "a": "aaa bbb ddd ccc if is over", "b":  
["eee fff ggg"], "c": {"d": "hhh iii"}, "d": "if", "e": "make  
chance Jenya pls", "f": "Dima, compressor is not  
working!!!" }'::jsonb);
```

В этом шаге значения ключей jsonb были нормализованы и отсортированы. Стоп-слова "if is over if" (предлоги, артикли, окончания) были исключены. Лексемы преобразованы в нижний регистр.

Вывод в терминал:

```
to_tsvector2  
-----  
'aaa':1 'bbb':2 'ccc':4 'ddd':3 'eee':9 'fff':10 'ggg':11  
'hhh':13 'iii':14
```

Выполнить преобразования JSONB в tsvector2:

```
SELECT to_tsvector2('{ "key": "value", "text": "This is a  
test." }'::jsonb) AS result;
```

В этом шаге значения ключей jsonb были нормализованы и отсортированы. Стоп-слова (предлоги, артикли, окончания) были исключены.

```
result  
-----  
'test':6 'valu':1 (1 row)
```

Выполнить преобразования JSON в tsvector:

```
SELECT jsonb_to_tsvector2('{ "key": "value", "text": "This is a  
test." }', 'all') AS result;
```

В этом шаге значения ключей jsonb были нормализованы и отсортированы при помощи функции jsonb_to_tsvector2(). Стоп-слова (предлоги, артикли, окончания) были исключены.


```
result
-----
'test':6 'valu':1 (1 row)
```

12.1.5. Функция «array_to_tsvector2». Преобразования из массива в строку

Выполнить преобразование массива в tsvector:

```
SELECT array_to_tsvector2(ARRAY['This', 'is', 'a', 'test']) AS
result;
```

содержащий лексемы из массива строк

```
result
-----
'This' 'a' 'is' 'test' (1 row)
```

12.1.6. Функция «tsvector2_to_array». Преобразование из строки в массив

Выполнить преобразование из строки в массив tsvector:

```
SELECT tsvector2_to_array(to_tsvector2('This is a test
document.')) AS result;
```

Возвращается массив строк с лексемами из tsvector2

Вывод в терминале:

```
result
-----
{document,test} (1 row)
```

12.1.7. Функция «tsvector2_stat». Получение статистики по лексемам

Создать таблицу:

```
CREATE TABLE doc ( id SERIAL PRIMARY KEY, content TEXT,
tsv_content TSVECTOR2 GENERATED ALWAYS AS
(to_tsvector2(content)) STORED );
```

Внести данные:

```
INSERT INTO doc (content) VALUES ('This is a test document.'),  
('Another example of a test document.'), ('PostgreSQL is great  
for full-text search.');
```

Использование `tsvector2_stat` для получения статистики:

```
SELECT * FROM tsvector2_stat('SELECT tsv_content FROM doc');
```

Вывод статистике в таблице где:

- `word` - лексема (слово);
- `ndoc` - количество документов где оно встречается;
- `nentry` - количество встреч в данном документе.

word	ndoc	nentry
text	1	1
test	2	2
search	1	1
postgresql	1	1
great	1	1
full-text	1	1
full	1	1
exampl	1	1
document	2	2
anoth	1	1

(10 rows)

12.1.8. Функции «`tsvector2_update_trigger`» и «`tsvector2_update_trigger_column`»

Создать таблицу:

```
CREATE TABLE messages ( id SERIAL PRIMARY KEY, title TEXT, body  
TEXT, tsv_content TSVECTOR2 GENERATED ALWAYS AS  
(to_tsvector2(title || ' ' || body)) STORED );
```

Создать триггер `tsvector2_update_trigger`:

```
CREATE TRIGGER tsvector2_update_trigger BEFORE INSERT OR UPDATE  
ON messages FOR EACH ROW EXECUTE PROCEDURE
```

```
tsvector2_update_trigger('tsv_content', 'pg_catalog.english',  
'title', 'body');
```

Вставить 2 строки:

```
INSERT INTO messages (title, body) VALUES ('First message',  
'This is the body of the first message.'), ('Second message',  
'This is the body of the second message.');
```

Проверить вставленные данные:

```
SELECT * FROM messages;
```

Обновить одну запись:

```
UPDATE messages SET body = 'Updated body of the first message.'  
WHERE id = 1;
```

Запросить данные:

```
SELECT tsv_content FROM messages WHERE id = 1;
```

Создать новую таблицу:

```
CREATE TABLE messages2 (  
    id SERIAL PRIMARY KEY,  
    tsv tsvector2,  
    body TEXT,  
    lang regconfig  
);
```

Создать триггер для апдейта колонки:

```
CREATE TRIGGER tsvector2_update_trigger_column BEFORE INSERT OR  
UPDATE ON messages2 FOR EACH ROW EXECUTE PROCEDURE  
tsvector2_update_trigger_column('tsv', 'lang', 'body');
```

Произвести вставку в таблицу:

```
INSERT INTO messages2 (body, lang) VALUES ('First message This  
is the body of the third message.', 'pg_catalog.english'),  
('Second message This is the body of the third message.',
```

```
'pg_catalog.english'), ('Third message This is the body of the  
third message.', 'pg_catalog.english');
```

Обновить одну из записей:

```
UPDATE messages2 SET body = 'Updated title for third message.'  
WHERE id = 3;
```

Произвести запрос:

```
SELECT tsv FROM messages2 WHERE id = 3;
```

Поле tsv обновилось после изменения.

Вывод в терминал:

```
tsv  
-----  
'messag':5 'third':4 'titl':2 'updat':1
```

13. КОМПОНЕНТ RUM

Компонент предоставляет метод доступа RUM для работы с индексами, основанный на коде методов доступа GIN.

Установка пакета компонента описана в документе «Руководство по установке».

Расширение компонента устанавливается от имени и с правами привилегированного пользователя SQL-командой:

```
CREATE EXTENSION rum;
```

Индекс GIN позволяет выполнять быстрый полнотекстовый поиск, используя типы tsvector и tsquery. Однако при этом применении он имеет следующие недостатки:

Компонент RUM сохраняет информацию о позиции лексем или метки времени.

13.1. Общие операторы

Операторы компонента перечислены в таблице 13.1.

Таблица 13.1 – Операторы компонента rum

Оператор	Возвращает	Описание
tsvector <=> tsquery	float4	Возвращает расстояние между значениями tsvector и tsquery.
timestamp <=> timestamp	float8	Возвращает расстояние между двумя значениями timestamp.
timestamp <= timestamp	float8	Возвращает расстояние только для возрастающих значений timestamp.
timestamp => timestamp	float8	Возвращает расстояние только для убывающих значений timestamp.

13.2. Классы операторов

Расширение rum предоставляет следующие классы операторов:

– rum_tsvector_ops

Сохраняет лексемы tsvector с информацией о позициях. Поддерживает упорядочивание с оператором <=> и поиск по префиксу.

– rum_tsvector_hash_ops

Сохраняет хеш лексем tsvector с информацией о позициях. Поддерживает упорядочивание с оператором <=>, и не поддерживает поиск по префиксу.

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

– rum_tsvector_addon_ops

Сохраняет лексемы tsvector с дополнительными данными любых типов, которые принимает RUM.

– rum_tsvector_hash_addon_ops

Сохраняет лексемы tsvector с дополнительными данными любых типов, которые принимает RUM. Не поддерживает поиск по префиксу.

– rum_tsquery_ops

Сохраняет ветви дерева запроса в дополнительной информации.

– rum_anyarray_ops

Сохраняет элементы массива anyarray и длину массива. Поддерживает упорядочивание с оператором <=>.

Индексируемые операторы: && @> <@ = %

– rum_anyarray_addon_ops

Сохраняет элементы anyarray с дополнительными данными любых типов, которые принимает RUM.

– rum_type_ops

Сохраняет лексемы соответствующего типа с информацией о позициях. В качестве типа в имени класса должно подставляться одно из следующих имён типов: int2, int4, int8, float4, float8, money, oid, timestamp, timestamptz, time, timetz, date, interval, macaddr, inet, cidr, text, varchar, char, bytea, bit, varbit, numeric.

Класс операторов rum_type_ops поддерживает упорядочивание с операторами <=>, <=| и |=>. Его можно использовать совместно с классами операторов rum_tsvector_addon_ops, rum_tsvector_hash_addon_ops и rum_anyarray_addon_ops.

Поддержка индексируемых операторов зависит от типа данных:

– Операторы < <= = >= > <=> <=| |=> поддерживаются для типов int2, int4, int8, float4, float8, money, oid, timestamp, timestamptz.

– Операторы < <= = >= > поддерживаются для типов time, timetz, date, interval, macaddr, inet, cidr, text, varchar, char, bytea, bit, varbit, numeric.

13.3. Примеры использования

13.3.1. Оператор rum_tsvector_ops

Создать таблицу:

```
CREATE TABLE test_rum(t text, a tsvector);
```

Создать триггер:

```
CREATE TRIGGER tsvectorupdate BEFORE UPDATE OR INSERT ON  
test_rum FOR EACH ROW EXECUTE PROCEDURE  
tsvector_update_trigger('a', 'pg_catalog.english', 't');
```

Заполнить таблицу:

```
INSERT INTO test_rum(t) VALUES ('The situation is most  
beautiful'); INSERT INTO test_rum(t) VALUES ('It is a  
beautiful');  
  
INSERT INTO test_rum(t) VALUES ('It looks like a beautiful  
place');
```

Создать индекс:

```
CREATE INDEX rumidx ON test_rum USING rum (a rum_tsvector_ops);
```

Выполнить чтение из таблицы:

```
SELECT t, a <=> to_tsquery('english', 'beautiful | place') AS  
rank FROM test_rum WHERE a @@ to_tsquery('english', 'beautiful  
| place') ORDER BY a <=> to_tsquery('english', 'beautiful |  
place');
```

Еще раз выполнить чтение из таблицы:

```
SELECT t, a <=> to_tsquery('english', 'place | situation') AS  
rank FROM test_rum WHERE a @@ to_tsquery('english', 'place |  
situation') ORDER BY a <=> to_tsquery('english', 'place |  
situation');
```

13.3.2. Оператор `rum_anyarray_ops`

Создать таблицу:

```
CREATE TABLE test_array (i int2[]);
```

Заполнить таблицу:

```
INSERT INTO test_array VALUES ('{}'), ('{0}'), ('{1,2,3,4}'),  
('{1,2,3}'), ('{1,2}'), ('{1}');
```

Создать индекс:

```
CREATE INDEX idx_array ON test_array USING rum (i  
rum_anyarray_ops);
```

Задать значение параметра:

```
SET enable_seqscan TO off;
```

Вывести план запроса:

```
EXPLAIN (COSTS OFF) SELECT * FROM test_array WHERE i && '{1}'  
ORDER BY i <=> '{1}' ASC;
```

Выполнить чтение из таблицы:

```
SELECT * FROM test_array WHERE i && '{1}' ORDER BY i <=> '{1}'  
ASC;
```


14. КОМПОНЕНТ XID64

Для обеспечения работы механизма MVCC (управление параллельным доступом посредством многоверсионности транзакций) СУБД «Jatoba» отслеживает, какие транзакции уже завершены, а какие еще активны.

Для этого каждой транзакции в СУБД «Jatoba» присваивается уникальный номер — идентификатор (xid). Его можно узнать, используя функцию txid_current().

Применяемый 64-битный счетчик транзакции xid64 в СУБД «Jatoba» в условиях высокой транзакционной нагрузки, позволяет избежать его «переполнения» (как в случае использования 32-битного счетчика и риска остановки работы БД) и дает гораздо большую свободу администраторам БД.

Компонент xid64 является частью СУБД «Jatoba» и включен по умолчанию, а также требует отдельной настройки параметров.

15. КОМПОНЕНТ HUNSPELL

Компонент «hunspell» - это свободная библиотека для проверки орфографии и морфологического анализа.

Компонент «hunspell» обеспечивает:

- Проверку правописания.
- Поддержку множества языков за счёт внешних словарей.

Основные особенности компонента «hunspell»:

- Формат словарей: использует .aff (аффиксы) и .dic (список слов) файлы.
- Поддержка Unicode: корректная работа с разными языками.
- Гибкость: позволяет добавлять пользовательские словари.

Расширение компонента «hunspell» устанавливается после установки пакета компонента, как описано в документе «Руководство по установке».

После установки пакета создаются следующие каталоги:

- /usr/jatoba-6/share/extension/hunspell_<lang>.control
- /usr/jatoba-6/share/extension/hunspell_<lang>--1.0.sql
- /usr/jatoba-6/share/tsearch_data/<lang>.affix
- /usr/jatoba-6/share/tsearch_data/<lang>.dict

Где <lang> – название поддерживаемого языка.

Список поддерживаемых компонентом языков:

- en_us – английский.
- ru_ru – русский.
- bg_bg – болгарский.
- de_de – немецкий.
- el_gr – греческий.
- et_ee – эстонский.

- fi_fi – финский.
- lt – литовский.
- lv_lv – латышский.
- pl_pl – польский.

После настройки расширения hunspell в СУБД «Jatoba» можно использовать морфологически правильный полнотекстовый поиск с поддержкой сложных словоформ. Это полезно для:

- Поиска с учетом падежей и склонений.
- Улучшения лингвистического анализа в приложениях.
- Исправления опечаток в текстовых запросах.

15.1. Установка расширения

В СУБД «Jatoba» поддержка морфологического анализа и проверки орфографии в полнотекстовом поиске (FTS) реализована через расширение hunspell.

Расширение компонента «hunspell» устанавливается от имени и с правами привилегированного пользователя SQL-командой:

```
CREATE EXTENSION hunspell_[lang];
```

Где lang – название поддерживаемого языка.

Пример установки расширения с поддержкой русского языка:

```
CREATE EXTENSION hunspell_ru_ru;
```

Проверка установки расширения в СУБД:

```
SELECT * FROM pg_available_extensions WHERE name =  
'hunspell_ru_ru';
```

Пример установки расширения с поддержкой английского языка:

```
CREATE EXTENSION hunspell_en_us;
```

Проверка установки расширения в СУБД:

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

```
SELECT * FROM pg_available_extensions WHERE name =  
'hunspell_en_us';
```

15.2. Добавление словаря

Создадим конфигурацию для русского языка:

```
CREATE TEXT SEARCH DICTIONARY russian_hunspell (  
    TEMPLATE = ispell,  
    DictFile = ru_ru,  
    AffFile = ru_ru,  
    StopWords = russian  
);
```

Здесь `russian_hunspell` – название создаваемого словаря.

Создадим текстовый поиск с использованием расширения «hunspell»:

```
CREATE TEXT SEARCH CONFIGURATION russian_fts (COPY = simple);  
ALTER TEXT SEARCH CONFIGURATION russian_fts  
    ALTER MAPPING FOR word, asciiword WITH russian_hunspell,  
    simple;
```

Проверка работы с использованием компонента `tsvector` (см. п.п. 12.1.3):

```
SELECT to_tsvector('russian_fts', 'пример текста для проверки  
орфографии');
```

Если словарь работает корректно, то слова будут нормализованы.

15.3. Взаимодействие с компонентом `tsvector2`

Компонент `hunspell` используется внутри FTS как словарь, который помогает правильно разбирать слова (например, учитывать падежи, склонения и т. д.).

Таким образом, компоненты `tsvector2` и `hunspell` позволяют:

- Искать слово в любом падеже (например, "проверка" → найдёт и "проверки", и "проверку").
- Игнорировать стоп-слова (если они заданы в конфигурации).

- Строить эффективные индексы для быстрого поиска.

Предположим, что создана таблица `test_table`.

```
CREATE TABLE test_table (  
    id INTEGER PRIMARY KEY GENERATED ALWAYS AS IDENTITY,  
    "text" text NULL,  
    "text_tsv" tsvector2 GENERATED ALWAYS AS  
to_tsvector2('russian_hunspell', text) NULL  
);
```

В таком случае запрос со словарями `hunspell` может выглядеть следующим образом:

```
SELECT id, text FROM test_table  
WHERE text_tsv @@ to_tsquery('russian_hunspell',  
'Запрашиваемый текст');
```

Компонент `hunspell` в СУБД «Jatoba» работает только в связке с FTS. Сам по себе он не проверяет орфографию "на лету" – он улучшает обработку текста для поиска.

15.4. Удаление расширения

Удаление расширения «`hunspell`» выполняется от имени и с правами привилегированного пользователя SQL-командой:

```
DROP EXTENSION hunspell_[lang];
```

Где `lang` – название поддерживаемого языка.

Если расширение «`hunspell`» используется в других объектах (например, в конфигурациях FTS), может потребоваться CASCADE:

```
DROP EXTENSION hunspell CASCADE;
```

Для проверки успешного удаления расширения «`hunspell`» необходимо выполнить следующий запрос:

```
ELECT * FROM pg_extension WHERE extname = 'hunspell_[lang]';
```

Где `lang` – название поддерживаемого языка.

Если запрос не возвращает строк – расширение успешно удалено.

Если были созданы конфигурации текстового поиска с расширением «hunspell», их тоже можно удалить:

```
DROP TEXT SEARCH CONFIGURATION russian_fts;  
DROP TEXT SEARCH DICTIONARY russian_hunspell;
```

Удаление пакета компонента «hunspell» выполняется средствами управления программным обеспечением операционной системы.

15.5. Решение проблем

15.5.1. Словарь не найден

Необходимо убедиться в том, что файлы .aff и .dic лежат в /usr/jatoba-6/share/tsearch_data/ и доступны пользователю postgres.

Если файлы называются ru_ru.aff и ru_ru.dic, необходимо указать DictFile = ru_ru при создании конфигурации языка (см. п.п. 15.2).

15.5.2. Ошибка прав доступа

Проверьте, что пользователь postgres имеет доступ к каталогу с словарями компонента /usr/jatoba-6/share/tsearch_data/:

```
# chmod -R a+r /usr/jatoba-6/share/tsearch_data/;
```

16. АВТОНОМНЫЕ ТРАНЗАКЦИИ

Автономной транзакцией, называется такая транзакция, которая выполняется в рамках другой транзакции (основной), и фиксируется или отменяется независимо от её исхода.

Важной особенностью автономных транзакции является то, что данный тип транзакций не видит эффект действия основной транзакции, как в случае, если бы основная транзакция еще не была зафиксирована.

Основная транзакция может видеть эффект действия вложенных в неё автономных транзакций в зависимости от её уровня изоляции. Например, основная транзакция с уровнем изоляции Read Committed будет видеть все изменения, вносимые вложенными в нее автономными транзакциями. Однако если для основной транзакции выбран уровень Repeatable Read, результат действия автономных транзакций для нее не будет виден.

При выполнении автономных транзакций в рамках одного сеанса работы могут возникать ситуации взаимной блокировки. Это следствие возникающего конфликта с другими приостановленными транзакциями в других сеансах работы. Также взаимная блокировка может возникать в случае попытки автономной транзакции получить доступ к ресурсам, которые уже заблокированы основной транзакцией. При обнаружении тех или иных блокировок СУБД «Jatoba» будет разрешать конфликты между блокировками согласно общим правилам работы транзакций. При обнаружении взаимной блокировки между автономной транзакцией и ее основной транзакции автономная транзакция будет прервана по ошибке взаимной блокировки.

СУБД «Jatoba» в рамках автономных транзакциях поддерживают только базовые SQL-команды: SELECT/INSERT/UPDATE/DELETE. Работа других SQL-команд не гарантируется.

SQL-выражения, в случае выполнения в автономной транзакции, фиксируют (COMMIT) или откатывают (ROLL BACK) без фиксации или отката основной транзакции.

В СУБД «Jatoba» транзакционные операторы (такие как например BEGIN, START TRANSACTION, COMMIT, END и ROLLBACK, ABORT) дополнены необязательным ключевым словом AUTONOMOUS.

Например:

```
BEGIN [AUTONOMOUS] [TRANSACTION] [ISOLATION LEVEL]  
END [AUTONOMOUS] [TRANSACTION]
```

Где ISOLATION LEVEL – это уровень изоляции автономной транзакции, может быть только READ COMMITTED.

Ключевое слово AUTONOMOUS применяется в том случае, если необходимо начать автономную транзакцию. При этом при завершении автономной транзакции ключевое слово AUTONOMOUS может не использоваться.

17. СООБЩЕНИЯ ОБ ОШИБКАХ

СУБД «Jatoba» использует пятисимвольные коды ошибок, описанные в стандарте SQL ISO/IEC 9075 «Язык баз данных SQL».

Перечень данных ошибок приведен в таблице 17.1.

Таблица 17.1 – Коды ошибок СУБД «Jatoba» по стандарту SQL ISO/IEC 9075

Код ошибки	Имя условия
Класс 00	Успешное завершение
00000	successful_completion
Класс 01	Предупреждение
01000	warning
0100C	dynamic_result_sets_returned
01008	implicit_zero_bit_padding
01003	null_value_eliminated_in_set_function
01007	privilege_not_granted
01006	privilege_not_revoked
01004	string_data_right_truncation
01P01	deprecated_feature
Класс 02	Нет данных (это также класс предупреждений согласно стандарту SQL)
02000	no_data
02001	no_additional_dynamic_result_sets_returned
Класс 03	SQL-оператор еще не завершен
03000	sql_statement_not_yet_complete
Класс 08	Исключение, связанное с подключением
08000	connection_exception
08003	connection_does_not_exist
08006	connection_failure
08001	sqlclient_unable_to_establish_sqlconnection
08004	sqlserver_rejected_establishment_of_sqlconnection
08007	transaction_resolution_unknown
08P01	protocol_violation
Класс 09	Исключение с действием триггера
09000	triggered_action_exception
Класс 0A	Неподдерживаемая функциональность
0A000	feature_not_supported
Класс 0B	Неверное начало транзакции
0B000	invalid_transaction_initiation
Класс 0F	Исключение с указателем на данные
0F000	locator_exception
0F001	invalid_locator_specification
Класс 0L	Неверный праводатель
0L000	invalid_grantor
0LP01	invalid_grant_operation
Класс 0P	Неверное указание роли
<div> <div>№ изменения: _____</div> <div>Подпись отв. лица: _____</div> <div>Дата внесения изм: _____</div> </div>	

Код ошибки	Имя условия
0P000	invalid_role_specification
Класс 0Z	Исключение диагностики
0Z000	diagnostics_exception
0Z002	stacked_diagnostics_accessed_without_active_handler
Класс 20	Case не найден
20000	case_not_found
Класс 21	Нарушение количества
21000	cardinality_violation
Класс 22	Исключение в данных
22000	data_exception
2202E	array_subscript_error
22021	character_not_in_repertoire
22008	datetime_field_overflow
22012	division_by_zero
22005	error_in_assignment
2200B	escape_character_conflict
22022	indicator_overflow
22015	interval_field_overflow
2201E	invalid_argument_for_logarithm
22014	invalid_argument_for_ntile_function
22016	invalid_argument_for_nth_value_function
2201F	invalid_argument_for_power_function
2201G	invalid_argument_for_width_bucket_function
22018	invalid_character_value_for_cast
22007	invalid_datetime_format
22019	invalid_escape_character
2200D	invalid_escape_octet
22025	invalid_escape_sequence
22P06	nonstandard_use_of_escape_character
22010	invalid_indicator_parameter_value
22023	invalid_parameter_value
2201B	invalid_regular_expression
2201W	invalid_row_count_in_limit_clause
2201X	invalid_row_count_in_result_offset_clause
2202H	invalid_tablesample_argument
2202G	invalid_tablesample_repeat
22009	invalid_time_zone_displacement_value
2200C	invalid_use_of_escape_character
2200G	most_specific_type_mismatch
22004	null_value_not_allowed
22002	null_value_no_indicator_parameter
22003	numeric_value_out_of_range
22026	string_data_length_mismatch
2001	string_data_right_truncation
22011	substring_error
22027	trim_error

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

Код ошибки	Имя условия
22024	unterminated_c_string
2200F	zero_length_character_string
22P01	floating_point_exception
22P02	invalid_text_representation
22P03	invalid_binary_representation
22P04	bad_copy_file_format
22P05	untranslatable_character
2200L	not_an_xml_document
2200M	invalid_xml_document
2200N	invalid_xml_content
2200S	invalid_xml_comment
2200T	invalid_xml_processing_instruction
Класс 23	Нарушение ограничения целостности
23000	integrity_constraint_violation
23001	restrict_violation
23502	not_null_violation
23503	foreign_key_violation
23505	unique_violation
23514	check_violation
23P01	exclusion_violation
Класс 24	Неверное состояние курсора
24000	invalid_cursor_state
Класс 25	Неверное состояние транзакции
25000	invalid_transaction_state
25001	active_sql_transaction
25002	branch_transaction_already_active
25008	held_cursor_requires_same_isolation_level
25003	inappropriate_access_mode_for_branch_transaction
25004	inappropriate_isolation_level_for_branch_transaction
25005	no_active_sql_transaction_for_branch_transaction
25006	read_only_sql_transaction
25007	schema_and_data_statement_mixing_not_supported
25P01	no_active_sql_transaction
25P02	in_failed_sql_transaction
Класс 26	Неверное имя SQL-оператора
26000	invalid_sql_statement_name
Класс 27	Нарушение при изменении данных в триггере
27000	triggered_data_change_violation
Класс 28	Неверное указание авторизации
28000	invalid_authorization_specification
28P01	invalid_password
Класс 2B	Зависимые описания привилегий все еще существуют
2B000	dependent_privilege_descriptors_still_exist
2BP01	dependent_objects_still_exist
Класс 2D	Неверное завершение транзакции
2D000	invalid_transaction_termination

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

Код ошибки	Имя условия
Класс 2F	Исключение в подпрограмме SQL
2F000	sql_routine_exception
2F005	function_executed_no_return_statement
2F002	modifying_sql_data_not_permitted
2F003	prohibited_sql_statement_attempted
2F004	reading_sql_data_not_permitted
Класс 34	Неверное имя курсора
34000	invalid_cursor_name
Класс 38	Исключение во внешней подпрограмме
38000	external_routine_exception
38001	containing_sql_not_permitted
38002	modifying_sql_data_not_permitted
38003	prohibited_sql_statement_attempted
38004	reading_sql_data_not_permitted
Класс 39	Исключение при вызове внешней подпрограммы
39001	invalid_sqlstate_returned
39000	external_routine_invocation_exception
39004	null_value_not_allowed
39P01	trigger_protocol_violated
39P02	srf_protocol_violated
39P03	event_trigger_protocol_violated
Класс 3B	Исключение точки сохранения
3B000	savepoint_exception
3B001	invalid_savepoint_specification
Класс 3D	Неверное имя каталога
3D000	invalid_catalog_name
Класс 3F	Неверное имя схемы
3F000	invalid_schema_name
Класс 40	Откат транзакции
40000	transaction_rollback
40002	transaction_integrity_constraint_violation
40001	serialization_failure
40003	statement_completion_unknown
40P01	deadlock_detected
Класс 42	Ошибка синтаксиса или нарушение правила доступа
42000	syntax_error_or_access_rule_violation
42601	syntax_error
42501	insufficient_privilege
42846	cannot_coerce
42803	grouping_error
42P20	windowing_error
42P19	invalid_recursion
42830	invalid_foreign_key
42602	invalid_name
42622	name_too_long
42939	reserved_name

Код ошибки	Имя условия
42804	datatype_mismatch
42P18	indeterminate_datatype
42P21	collation_mismatch
42P22	indeterminate_collation
42809	wrong_object_type
42703	undefined_column
42883	undefined_function
42P01	undefined_table
42P02	undefined_parameter
42704	undefined_object
42701	duplicate_column
42P03	duplicate_cursor
42P04	duplicate_database
42723	duplicate_function
42P05	duplicate_prepared_statement
42P06	duplicate_schema
42P07	duplicate_table
42712	duplicate_alias
42710	duplicate_object
42702	ambiguous_column
42725	ambiguous_function
42P08	ambiguous_parameter
42P09	ambiguous_alias
42P10	invalid_column_reference
42611	invalid_column_definition
42P11	invalid_cursor_definition
42P12	invalid_database_definition
42P13	invalid_function_definition
42P14	invalid_prepared_statement_definition
42P15	invalid_schema_definition
42P16	invalid_table_definition
42P17	invalid_object_definition
Класс 44	Нарушение WITH CHECK OPTION
44000	with_check_option_violation
Класс 53	Нехватка ресурсов
53000	insufficient_resources
53100	disk_full
53200	out_of_memory
53300	too_many_connections
53400	configuration_limit_exceeded
Класс 54	Превышение ограничения программы
54000	program_limit_exceeded
54001	statement_too_complex
54011	too_many_columns
54023	too_many_arguments
Класс 55	Объект не в требуемом состоянии

Код ошибки	Имя условия
55000	object_not_in_prerequisite_state
55006	object_in_use
55P02	cant_change_runtime_param
55P03	lock_not_available
Класс 57	Вмешательство оператора
57000	operator_intervention
57014	query_canceled
57P01	admin_shutdown
57P02	crash_shutdown
57P03	cannot_connect_now
57P04	database_dropped
Класс 58	Ошибка системы
58000	system_error
58030	io_error
58P01	undefined_file
58P02	duplicate_file
Класс F0	Ошибка файла конфигурации
F0000	config_file_error
F0001	lock_file_exists
Класс HV	Ошибка обертки сторонних данных (SQL/MED)
HV000	fdw_error
HV005	fdw_column_name_not_found
HV002	fdw_dynamic_parameter_value_needed
HV010	fdw_function_sequence_error
HV021	fdw_inconsistent_descriptor_information
HV024	fdw_invalid_attribute_value
HV007	fdw_invalid_column_name
HV008	fdw_invalid_column_number
HV004	fdw_invalid_data_type
HV006	fdw_invalid_data_type_descriptors
HV091	fdw_invalid_descriptor_field_identifier
HV00B	fdw_invalid_handle
HV00C	fdw_invalid_option_index
HV00D	fdw_invalid_option_name
HV090	fdw_invalid_string_length_or_buffer_length
HV00A	fdw_invalid_string_format
HV009	fdw_invalid_use_of_null_pointer
HV014	fdw_too_many_handles
HV001	fdw_out_of_memory
HV00P	fdw_no_schemas
HV00J	fdw_option_name_not_found
HV00K	fdw_reply_handle
HV00Q	fdw_schema_not_found
HV00R	fdw_table_not_found
HV00L	fdw_unable_to_create_execution
HV00M	fdw_unable_to_create_reply

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

Код ошибки	Имя условия
HV00N	fdw_unable_to_establish_connection
Класс P0	Ошибка PL/pgSQL
P0000	plpgsql_error
P0001	raise_exception
P0002	no_data_found
P0003	too_many_rows
P0004	assert_failure
Класс XX	Внутренняя ошибка
XX000	internal_error
XX001	data_corrupted
XX002	index_corrupted

Возможные сообщения об ошибках, связанные с действиями по выполнению функций безопасности, приведены в таблице 17.2.

Таблица 17.2 – Перечень ошибок при выполнении ФБО

Мера	Код	Сообщение	Сообщение в транскрипции
(ИАФ)	c:43	msgid "could not look up effective user ID %ld: %s"	msgstr "выяснить эффективный идентификатор пользователя (%ld) не удалось: %s"
(ИАФ)	c:554	msgid "user does not exist"	msgstr "пользователь не существует"
(ИАФ)	c:60	msgid "user name lookup failure: error code %lu"	msgstr "распознать имя пользователя не удалось (код ошибки: %lu)"
(ИАФ)	c:1796	msgid "Enter new password: "	msgstr "Введите новый пароль: "
(ИАФ)	c:1801	msgid "Passwords didn't match.\n"	msgstr "Пароли не совпадают.\n"
(ИАФ)	c:3329	msgid "Cannot login"	msgstr "Вход запрещен"
(ИАФ)	c:3358	msgid "Password valid until "	msgstr "Пароль действует до "
(ИАФ)	c:140	msgid " -w, --no-password never prompt for password\n"	msgstr " -w, --no-password не запрашивать пароль\n"
(ИАФ)	c:141	msgid " -W, --password force password prompt (should happen \"automatically)\n"	msgstr " -W, --password запрашивать пароль всегда (обычно не требуется)\n"
(ИАФ)	c:297	msgid " \\password [USERNAME] securely change the password for a user\n"	msgstr " \\password [ИМЯ] безопасно сменить пароль пользователя\n"
(ИАФ)	c:1093	msgid "password too long"	msgstr "слишком длинный пароль"
(ИАФ)	c:249	msgid "client selected an invalid SASL authentication mechanism"	msgstr "клиент выбрал неверный механизм аутентификации SASL"
(ИАФ)	c:296	msgid "password authentication failed for user \"%s\""	msgstr "пользователь \"%s\" не прошел проверку подлинности (по паролю)"
(ИАФ)	c:322	msgid "authentication failed for user \"%s\": invalid authentication method"	msgstr "пользователь \"%s\" не прошел проверку подлинности: неверный метод проверки"
(ИАФ)	c:3282	msgid "could not perform MD5 encryption of received packet"	msgstr "не удалось вычислить MD5 для принятого пакета"
(ИАФ)	c:80	msgid "User \"%s\" has an expired password."	msgstr "Срок пароля пользователя \"%s\" истек."

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

Мера	Код	Сообщение	Сообщение в транскрипции
(ИАФ)	c:182	msgid "User \"%s\" has a password that cannot be used with MD5 authentication."	msgstr "Пользователь \"%s\" имеет пароль, неподходящий для аутентификации по MD5."
(ИАФ)	c:206 c:247 c:271	msgid "Password does not match for user \"%s\"."	msgstr "Пароль не подходит для пользователя \"%s\"."
(ИАФ)	c:290	msgid "Password of user \"%s\" is in unrecognized format."	msgstr "Пароль пользователя \"%s\" представлен в неизвестном формате."
(ИАФ)	c:408	msgid "%s: real and effective user IDs must match\n"	msgstr "%s: фактический и эффективный ID пользователя должны совпадать\n"
(ИАФ)	c:2397	msgid "Sets the maximum allowed time to complete client authentication."	msgstr "Ограничивает время, за которое клиент должен пройти аутентификацию."
(ИАФ)	y:1026	msgid "UNENCRYPTED PASSWORD is no longer supported"	msgstr "вариант UNENCRYPTED PASSWORD более не поддерживается"
(ИАФ)	y:1027	msgid "Remove UNENCRYPTED to store the password in encrypted form instead."	msgstr "Удалите слово UNENCRYPTED, чтобы сохранить пароль в зашифрованном виде."
(ИАФ)	l:1556	#~ msgid "%s: could not determine user name (GetUserName failed)\n"	#~ msgstr "%s: не удалось определить имя пользователя (ошибка в GetUserName)\n"
(ИАФ)	l:1556	#~ msgid "User \"%s\" has an empty password."	#~ msgstr "У пользователя \"%s\" пустой пароль."
(УПД)	c:45	msgid "command not executable"	msgstr "неисполняемая команда"
(УПД)	c:552	msgid "could not get home directory for user ID %ld: %s\n"	msgstr "не удалось получить домашний каталог пользователя с ид. %ld: %s\n"
(УПД)	c:1098	msgid "The server (version %s) does not support altering default privileges.\n"	msgstr "Сервер (версия %s) не поддерживает изменение прав по умолчанию.\n"
(УПД)	c:3323	msgid "Create role"	msgstr "Создает роли"
(УПД)	c:3326	msgid "Create DB"	msgstr "Создает БД"
(УПД)	c:3408	msgid "The server (version %s) does not support per-database role settings.\n"	"Сервер (версия %s) не поддерживает назначение параметров ролей для баз "
(УПД)	c:3444	msgid "Did not find any settings for role \"%s\" and database \"%s\".\n"	msgstr "Параметры для роли \"%s\" и базы данных \"%s\" не найдены.\n"
(УПД)	c:3447	msgid "Did not find any settings for role \"%s\".\n"	msgstr "Параметры для роли \"%s\" не найдены.\n"
(УПД)	c:3450	msgid "Did not find any settings.\n"	msgstr "Никакие параметры не найдены.\n"
(УПД)	c:3364	msgid "permission denied for aggregate %s"	msgstr "нет доступа к агрегату %s"
(УПД)	c:3367	msgid "permission denied for collation %s"	msgstr "нет доступа к правилу сортировки %s"
(УПД)	c:3370	msgid "permission denied for column %s"	msgstr "нет доступа к столбцу %s"
(УПД)	c:3373	msgid "permission denied for conversion %s"	msgstr "нет доступа к преобразованию %s"

Мера	Код	Сообщение	Сообщение в транскрипции
(УПД)	c:3376	msgid "permission denied for database %s"	msgstr "нет доступа к базе данных %s"
(УПД)	c:3379	msgid "permission denied for domain %s"	msgstr "нет доступа к домену %s"
(УПД)	c:3382	msgid "permission denied for event trigger %s"	msgstr "нет доступа к событийному триггеру %s"
(УПД)	c:3385	msgid "permission denied for extension %s"	msgstr "нет доступа к расширению %s"
(УПД)	c:3388	msgid "permission denied for foreign-data wrapper %s"	msgstr "нет доступа к обертке сторонних данных %s"
(УПД)	c:3391	msgid "permission denied for foreign server %s"	msgstr "нет доступа к стороннему серверу %s"
(УПД)	c:3394	msgid "permission denied for foreign table %s"	msgstr "нет доступа к сторонней таблице %s"
(УПД)	c:3397	msgid "permission denied for function %s"	msgstr "нет доступа к функции %s"
(УПД)	c:3400	msgid "permission denied for index %s"	msgstr "нет доступа к индексу %s"
(УПД)	c:3403	msgid "permission denied for language %s"	msgstr "нет доступа к языку %s"
(УПД)	c:3406	msgid "permission denied for large object %s"	msgstr "нет доступа к большому объекту %s"
(УПД)	c:3409	msgid "permission denied for materialized view %s"	msgstr "нет доступа к материализованному представлению %s"
(УПД)	c:3412	msgid "permission denied for operator class %s"	msgstr "нет доступа к классу операторов %s"
(УПД)	c:3415	msgid "permission denied for operator %s"	msgstr "нет доступа к оператору %s"
(УПД)	c:3418	msgid "permission denied for operator family %s"	msgstr "нет доступа к семейству операторов %s"
(УПД)	c:3421	msgid "permission denied for policy %s"	msgstr "нет доступа к политике %s"
(УПД)	c:3424	msgid "permission denied for procedure %s"	msgstr "нет доступа к процедуре %s"
(УПД)	c:3427	msgid "permission denied for publication %s"	msgstr "нет доступа к публикации %s"
(УПД)	c:3430	msgid "permission denied for routine %s"	msgstr "нет доступа к подпрограмме %s"
(УПД)	c:3433	msgid "permission denied for schema %s"	msgstr "нет доступа к схеме %s"
(УПД)	c:1852	msgid "permission denied for sequence %s"	msgstr "нет доступа к последовательности %s"
(УПД)	c:3439	msgid "permission denied for statistics object %s"	msgstr "нет доступа к объекту статистики %s"
(УПД)	c:3442	msgid "permission denied for subscription %s"	msgstr "нет доступа к подписке %s"
(УПД)	c:3445	msgid "permission denied for table %s"	msgstr "нет доступа к таблице %s"

Мера	Код	Сообщение	Сообщение в транскрипции
(УПД)	c:3448	msgid "permission denied for tablespace %s"	msgstr "нет доступа к табличному пространству %s"
(УПД)	c:3451	msgid "permission denied for text search configuration %s"	msgstr "нет доступа к конфигурации текстового поиска %s"
(УПД)	c:3454	msgid "permission denied for text search dictionary %s"	msgstr "нет доступа к словарю текстового поиска %s"
(УПД)	c:3457	msgid "permission denied for type %s"	msgstr "нет доступа к типу %s"
(УПД)	c:3460	msgid "permission denied for view %s"	msgstr "нет доступа к представлению %s"
(УПД)	c:3643	msgid "permission denied for column \"%s\" of relation \"%s\""	msgstr "нет доступа к столбцу \"%s\" отношения \"%s\""
(УПД)	c:58	msgid "permission denied to create access method \"%s\""	msgstr "нет прав на создание метода доступа \"%s\""
(УПД)	c:60	msgid "Must be superuser to create an access method."	msgstr "Для создания метода доступа нужно быть суперпользователем."
(УПД)	c:68	msgid "access method \"%s\" already exists"	msgstr "метод доступа \"%s\" уже существует"
(УПД)	c:123	msgid "must be superuser to drop access methods"	msgstr "для удаления методов доступа нужно быть суперпользователем"
(УПД)	c:824	msgid "" "must be superuser or a member of the pg_write_server_files role to COPY to a ""file"	"для выполнения COPY с записью в файл нужно быть суперпользователем или ""членом роли pg_write_server_files"
(УПД)	c:311	msgid "permission denied to create database"	msgstr "нет прав на создание базы данных"
(УПД)	c:346	msgid "permission denied to copy database \"%s\""	msgstr "нет прав на копирование базы данных \"%s\""
(УПД)	c:480 c:1016	msgid "database \"%s\" already exists"	msgstr "база данных \"%s\" уже существует"
(УПД)	c:839	msgid "cannot drop a template database"	msgstr "удалить шаблон базы данных нельзя"
(УПД)	c:845	msgid "cannot drop the currently open database"	msgstr "удалить базу данных, открытую в данный момент, нельзя"
(УПД)	c:1007	msgid "permission denied to rename database"	msgstr "нет прав на переименование базы данных"
(УПД)	c:1667	msgid "permission denied to change owner of database"	msgstr "нет прав на изменение владельца базы данных"
(УПД)	c:620	msgid "permission denied to change owner of event trigger \"%s\""	msgstr "нет прав на изменение владельца событийного триггера \"%s\""
(УПД)	c:807	msgid "permission denied to create extension \"%s\""	msgstr "нет прав на создание расширения \"%s\""
(УПД)	c:813	msgid "permission denied to update extension \"%s\""	msgstr "нет прав на изменение расширения \"%s\""
(УПД)	c:696	msgid "permission denied to alter foreign-data wrapper \"%s\""	msgstr "нет прав на изменение обертки сторонних данных \"%s\""
(УПД)	c:924	msgid "permission denied: \"%s\" is a system catalog"	msgstr "доступ запрещен: \"%s\" - это системный каталог"

Мера	Код	Сообщение	Сообщение в транскрипции
(УПД)	c:170	msgid "ignoring specified roles other than PUBLIC"	msgstr "все указанные роли, кроме PUBLIC, игнорируются"
(УПД)	c:802 c:1247	msgid "policy \"%s\" for table \"%s\" already exists"	msgstr "политика \"%s\" для таблицы \"%s\" уже существует"
(УПД)	c:681	msgid "permission denied to change owner of publication \"%s\""	msgstr "нет прав на изменение владельца публикации \"%s\""
(УПД)	c:590	msgid "could not set permissions on directory \"%s\": %m"	msgstr "не удалось установить права для каталога \"%s\": %m"
(УПД)	c:295	msgid "must be superuser to create superusers"	msgstr "для создания суперпользователей нужно быть суперпользователем"
(УПД)	c:302	msgid "must be superuser to create replication users"	msgstr "для создания пользователей-репликаторов нужно быть суперпользователем"
(УПД)	c:309 c:707	msgid "must be superuser to change bypassrsls attribute"	msgstr "для изменения атрибута bypassrsls нужно быть суперпользователем"
(УПД)	c:316	msgid "permission denied to create role"	msgstr "нет прав для создания роли"
(УПД)	c:340 c:1210	msgid "role \"%s\" already exists"	msgstr "роль \"%s\" уже существует"
(УПД)	c:406 c:816	msgid "empty string is not a valid password, clearing password"	msgstr "пустая строка не является допустимым паролем; пароль сбрасывается"
(УПД)	c:1593	msgid "must be superuser to alter superusers"	msgstr "для модификации суперпользователей нужно быть суперпользователем"
(УПД)	c:700	msgid "must be superuser to alter replication users"	msgstr "для модификации пользователей-репликаторов нужно быть суперпользователем"
(УПД)	c:723 c:923	msgid "permission denied"	msgstr "нет доступа"
(УПД)	c:975	msgid "permission denied to drop role"	msgstr "нет прав для удаления роли"
(УПД)	c:1026 c:1030	msgid "current user cannot be dropped"	msgstr "пользователь не может удалить сам себя"
(УПД)	c:1034	msgid "session user cannot be dropped"	msgstr "пользователя текущего сеанса нельзя удалить"
(УПД)	c:1045	msgid "must be superuser to drop superusers"	msgstr "для удаления суперпользователей нужно быть суперпользователем"
(УПД)	c:1182	msgid "session user cannot be renamed"	msgstr "пользователя текущего сеанса нельзя переименовать"
(УПД)	c:1186	msgid "current user cannot be renamed"	msgstr "пользователь не может переименовать сам себя"
(УПД)	c:1220	msgid "must be superuser to rename superusers"	msgstr "для переименования суперпользователей нужно быть суперпользователем"
(УПД)	c:1227	msgid "permission denied to rename role"	msgstr "нет прав на переименование роли"

Мера	Код	Сообщение	Сообщение в транскрипции
(УПД)	c:1248	msgid "MD5 password cleared because of role rename"	msgstr "в результате переименования роли очищен MD5-хеш пароля"
(УПД)	c:1346	msgid "permission denied to drop objects"	msgstr "нет прав на удаление объектов"
(УПД)	c:1373 c:1382	msgid "permission denied to reassign objects"	msgstr "нет прав для переназначения объектов"
(УПД)	c:1457 c:1601	msgid "must have admin option on role \"%s\""	msgstr "требуется право admin для роли \"%s\""
(УПД)	c:1474	msgid "must be superuser to set grantor"	msgstr "для назначения права управления правами нужно быть суперпользователем"
(УПД)	c:466	msgid "could not access file \"%s\": %m"	msgstr "нет доступа к файлу \"%s\": %m"
(УПД)	c:847 c:296	msgid "permission denied for large object %u"	msgstr "нет доступа к большому объекту %u"
(УПД)	c:696	msgid "could not set permissions of file \"%s\": %m"	msgstr "не удалось установить права доступа для файла \"%s\": %m"
(УПД)	c:157	msgid "data directory \"%s\" has invalid permissions"	msgstr "для каталога данных \"%s\" установлены неправильные права доступа"
(УПД)	c:645	msgid "role \"%s\" is not permitted to log in"	msgstr "для роли \"%s\" вход запрещен"
(УПД)	c:663	msgid "too many connections for role \"%s\""	msgstr "слишком много подключений для роли \"%s\""
(УПД)	c:723	msgid "permission denied to set session authorization"	msgstr "нет прав для смены объекта авторизации в сеансе"
(УПД)	c:346	msgid "User does not have CONNECT privilege."	msgstr "Пользователь не имеет привилегии CONNECT."
(УПД)	c:363	msgid "too many connections for database \"%s\""	msgstr "слишком много подключений к БД \"%s\""
(УПД)	c:803	msgid "must be superuser or replication role to start walsender"	msgstr "для запуска процесса walsender требуется роль репликации или права суперпользователя "
(УПД)	y:1089	msgid "unrecognized role option \"%s\""	msgstr "нераспознанный параметр роли \"%s\""
(УПД)	y:14900 y:14907	msgid "%s cannot be used as a role name here"	msgstr "%s нельзя использовать здесь как имя роли"
(ОЦЛ)	c:3003	msgid "Cannot add header to table content: column count of %d exceeded.\n"	"Ошибка добавления заголовка таблицы: превышен предел числа столбцов (%d).\n"
(ОЦЛ)	c:3043	msgid "Cannot add cell to table content: total cell count of %d exceeded.\n"	"Ошибка добавления ячейки в таблицу: превышен предел числа ячеек (%d).\n"
(ОЦЛ)	c:3292	msgid "invalid output format (internal error): %d"	msgstr "неверный формат вывода (внутренняя ошибка): %d"
(ОЦЛ)	c:220	msgid "Invalid command \\%s. Try \\? for help.\n"	msgstr "Неверная команда \\%s. Справка по командам: \\?\n"
(ОЦЛ)	c:222	msgid "invalid command \\%s\n"	msgstr "неверная команда \\%s\n"
(ОЦЛ)	c:4496	msgid "invalid field size"	msgstr "неверный размер поля"

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

Мера	Код	Сообщение	Сообщение в транскрипции
(ОЦЛ)	c:4519	msgid "incorrect binary data format"	msgstr "неверный двоичный формат данных"
(ОЦЛ)	c:3209	msgid "invalid input string for \\"Y,YYY\\""	msgstr "ошибка синтаксиса в значении для шаблона \\"Y,YYY\\""
(ОЦЛ)	c:3724	msgid "hour \"%d\" is invalid for the 12-hour clock"	msgstr "час \"%d\" не соответствует 12-часовому формату времени"
(ОЦЛ)	c:2389	msgid "The given value did not match any of the allowed values for this field."	msgstr "Данное значение не соответствует ни одному из допустимых значений для этого \"поля.\""
(ОЦЛ)	c:2266	msgid "Value must be in the range %d to %d."	msgstr "Значение должно быть в интервале %d..%d."
(ОЦЛ)	c:2259	msgid "Value must be an integer."	msgstr "Значение должно быть целым числом."
(ОЦЛ)	c:2244 c:2257	msgid "invalid value \"%s\" for \"%s\\""	msgstr "неверное значение \"%s\" для \"%s\\""
(ОЦЛ)	c:2229	msgid "source string too short for \"%s\" formatting field"	msgstr "входная строка короче, чем требует поле форматирования \"%s\\""
(ОЦЛ)	c:1392	msgid "\"%s\" is not a number"	msgstr "\"%s\" не является числом"
(ОЦЛ)	c:488	msgid "invalid format specification for an interval value"	msgstr "неправильная спецификация формата для целого числа"
(ОЦЛ)	c:125 c:51 c:61	msgid "invalid input syntax for integer: \"%s\\""	msgstr "неверное значение для целого числа: \"%s\\""
(ОЦЛ)	l:1556	#~ msgid "access method name cannot be qualified"	#~ msgstr "имя метода доступа не может быть составным"
(ОЦЛ)	l:1556	#~ msgid "database name cannot be qualified"	#~ msgstr "имя базы данных не может быть составным"
(ОЦЛ)	l:1556	#~ msgid "extension name cannot be qualified"	#~ msgstr "имя расширения не может быть составным"
(ОЦЛ)	l:1556	#~ msgid "tablespace name cannot be qualified"	#~ msgstr "имя табличного пространства не может быть составным"
(ОЦЛ)	l:1556	#~ msgid "role name cannot be qualified"	#~ msgstr "имя роли не может быть составным"
(ОЦЛ)	l:1556	#~ msgid "schema name cannot be qualified"	#~ msgstr "имя схемы не может быть составным"
(ОЦЛ)	l:1556	#~ msgid "language name cannot be qualified"	#~ msgstr "имя языка не может быть составным"
(ОЦЛ)	l:1556	#~ msgid "foreign-data wrapper name cannot be qualified"	#~ msgstr "имя обертки сторонних данных не может быть составным"
(ОЦЛ)	l:1556	#~ msgid "server name cannot be qualified"	#~ msgstr "имя сервера не может быть составным"
(ОЦЛ)	l:1556	#~ msgid "event trigger name cannot be qualified"	#~ msgstr "имя событийного триггера не может быть составным"
(ОЦЛ)	l:1556	#~ msgid "invalid input syntax for type real: \"%s\\""	#~ msgstr "неверный синтаксис для типа real: \"%s\\""
(ОЦЛ)	l:1556	#~ msgid "invalid value for parameter \\"replication\\""	#~ msgstr "неверное значение параметра \\"replication\\""
(ОЦЛ)	l:1556	#~ msgid "invalid symbol"	#~ msgstr "неверный символ"

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

Мера	Код	Сообщение	Сообщение в транскрипции
(РСБ)	c:179	msgid "invalid data in history file \"%s\""	msgstr "неверные данные в файле истории \"%s\""
(РСБ)	c:915	msgid "%s: invalid datetoken tables, please fix\n"	msgstr "%s: ошибка в таблицах маркеров времени, требуется исправление\n"
(РСБ)	c:565 c:579	msgid "could not create pipe for syslog: %m"	msgstr "не удалось создать канал для syslog: %m"
(РСБ)	c:1102	msgid "could not write to log file: %s\n"	msgstr "не удалось записать в файл протокола: %s\n"
(РСБ)	c:1219	msgid "could not open log file \"%s\": %m"	msgstr "не удалось открыть файл протокола \"%s\": %m"
(РСБ)	c:630	msgid "could not fork system logger: %m"	msgstr "не удалось породить процесс системного протоколирования: %m"
(РСБ)	c:484	msgid "invalid input syntax for numeric time zone: \"%s\""	msgstr "неверный синтаксис для числового часового пояса: \"%s\""
(РСБ)	c:301	msgid "invalid time zone file name \"%s\""	msgstr "неправильное имя файла часовых поясов: \"%s\""

18. ДЕЙСТВИЯ ПОСЛЕ СБОЕВ И ОШИБОК ЭКСПЛУАТАЦИИ СУБД «ЯТОВА»

При эксплуатации СУБД «Jatoba» возможно возникновение следующих ошибок:

- сбой инициализация расширения «securityprofile» (п. 18.3);
- ошибки создания пользователя (п. 18.4);
- ошибки, возникающие при использовании профиля парольных политик (п. 18.5);
- ошибка авторизации (п. 18.6).

18.1. Временная блокировка пользователей СУБД и суперпользователя

Ошибка может возникнуть если не были выполнены требования п. 6.1.3.7 настоящего документа.

```
Account must have password. Account locked temporary.
```

В случае когда временно заблокированы УЗ пользователей СУБД, следует выполнить действия описанные в п. 6.2.4.

Если заблокирован пользователь «postgres», то порядок действий должен быть следующим:

- 1) Изменить метод аутентификации в конфигурационном файле «pg_hba.conf» на «TRUST».
- 2) В файле конфигурационном файле «postgresql.conf» **не отключать** параметр:

```
shared_preload_libraries = 'securityprofile'
```

- 3) Перезапустить СУБД «Jatoba» командами:

- в ОС Windows:

```
net stop JatobaServer  
net start JatobaServer
```

- в GNU Linux:

```
systemctl stop jatoba-<ver>
systemctl start jatoba-<ver>
systemctl status jatoba-<ver>
```

4) Войти в СУБД от имени и с правами пользователя «postgres» и изменить пароль в СУБД при помощи команды:

```
ALTER ROLE <имя учетной записи пользователя> password '<пароль пользователя>';
```

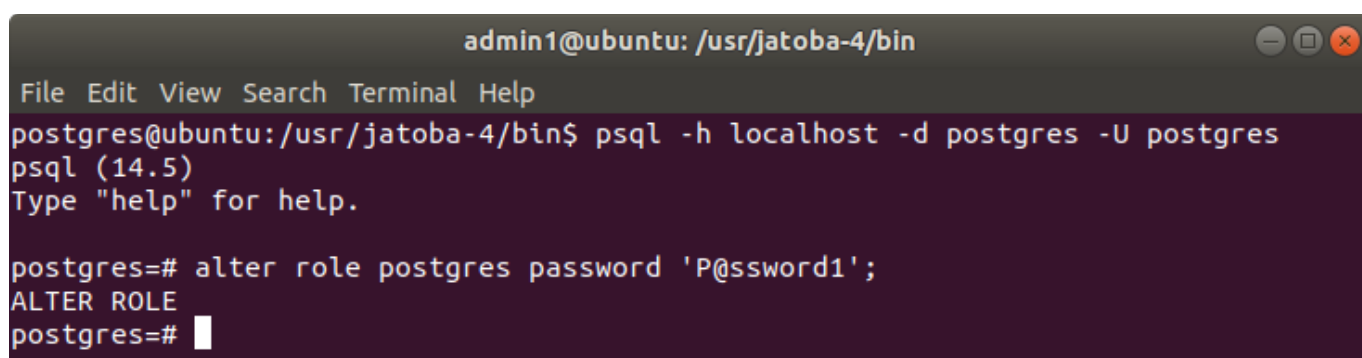


Рисунок 18.1 – Смена пароля пользователя «postgres»

5) Изменить метод аутентификации в конфигурационном файле «pg_hba.conf» на «md5» или другой метод парольной аутентификации;

6) Перезапустить СУБД «Jatoba» командами:

– в ОС Windows:

```
net stop JatobaServer
net start JatobaServer
```

– в GNU Linux:

```
systemctl stop jatoba-<ver>
systemctl start jatoba-<ver>
systemctl status jatoba-<ver>
```

7) Проверить работоспособность СУБД, войдя от имени и с правами пользователя «postgres».

18.2. Блокировка суперпользователя СУБД

Ошибка может возникнуть в случае:

- истечение срока действия пароля, без своевременного обновления пароля;
- превышение допустимого числа попыток ввода некорректного пароля;
- отсутствие подключений к серверу с использованием учетных данных, дольше разрешенного времени.

Если заблокирован пользователь «postgres», то порядок действий должен быть следующим:

- 1) Изменить метод аутентификации в конфигурационном файле «pg_hba.conf» на «TRUST».

Будет достаточно добавить строку с именем суперпользователя:

```
local all <имя роли суперпользователя> trust
```

- 2) В файле конфигурационном файле «postgresql.conf» **не отключать** параметр:

```
shared_preload_libraries = 'securityprofile'
```

- 3) Перезапустить СУБД «Jatoba» командами:

- в ОС Windows:

```
net stop JatobaServer  
net start JatobaServer
```

- в GNU Linux:

```
systemctl stop jatoba-<ver>  
systemctl start jatoba-<ver>  
systemctl status jatoba-<ver>
```

- 4) Выполнить команду разблокировки учетной записи заблокированного пользователя:

- SQL-командой:

```
SELECT  
securityprofile.unlock_account('<имя заблокированной роли>',  
0);
```

- Командой в терминале ОС:

```
postgres@host$ psql -p 5435 -d secprofdb -U pgadmin -c "SELECT  
securityprofile.unlock_account('<имя заблокированной роли>',  
0);"
```

5) Открыть в редакторе файл "pg_hba.conf" и восстановить режим аутентификации, отключив режим "trust".

6) Перезапустить службу СУБД.

```
SYSTEMCTL RESTART <Имя службы СУБД>
```

7) Проверить работоспособность СУБД, войдя от имени и с правами пользователя «postgres».

18.3. Сбой инициализация расширения «securityprofile»

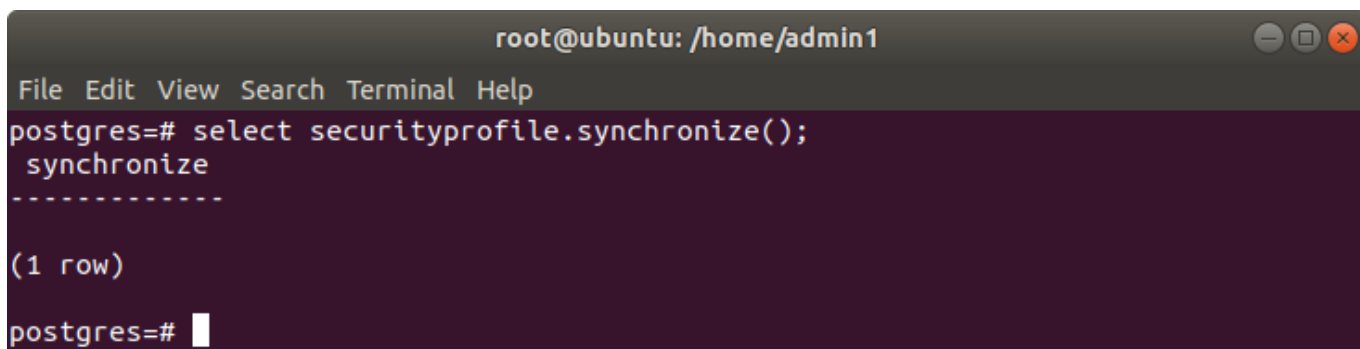
После перезагрузки сервера СУБД либо службы «JatobaServer» СУБД у пользователей, при авторизации может возникать ошибка:

```
«FATAL: Extension securityprofile need to be initialized by  
superuser.»
```

На процесс авторизации пользователей в СУБД данная ошибка не влияет.

Для устранения возникшей ошибки, следует повторно инициализировать расширение «SecurityProfile», выполнив команду:

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

A screenshot of a terminal window with a dark background. The title bar at the top reads 'root@ubuntu: /home/admin1'. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows a PostgreSQL prompt 'postgres=#' followed by the command 'select securityprofile.synchronize();'. Below the command, there is a dashed line and the text '(1 row)'. The prompt 'postgres=#' is followed by a cursor.

```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
postgres=# select securityprofile.synchronize();
synchronize
-----
(1 row)
postgres=#
```

Рисунок 18.2 – Команда выполнения инициализации расширения «SecurityProfile»

18.4. Ошибки создания пользователя

При попытке пользователя, обладающего привилегией создания ролей «Create roles», создать пользователя может возникнуть ошибка:

```
«permission denied for schema securityprofile»
```

Это означает, что у пользователя отсутствует доступ к схеме «securityprofile».

Для устранения ошибки следует предоставить права пользователю на использование схемы, выполнив команды:

```
GRANT CREATE ON SCHEMA securityprofile TO <имя пользователя>;
GRANT USAGE ON SCHEMA securityprofile TO <имя пользователя>;
```

После инициализации расширения «securityprofile» активируется парольная политика по умолчанию.

Пароль пользователя должен содержать:

- 1 символ в верхнем регистре;
- 1 символ в нижнем регистре;
- 1 спецсимвол.

18.5. Ошибки, возникающие при использовании профиля парольных политик «securityprofile»

Перечень ошибок, возникающих при использовании профиля парольных политик «securityprofile», приведен в таблице 18.1.

Таблица 18.1 – Перечень ошибок

№	Текст ошибки	Перевод	Пути исправления
1.	Schema (%s) does not exist. Probably extension (%s) is not created yet	Схема (<i>имя схемы</i>) не существует. Вероятно, расширение (<i>имя профиля</i>) еще не создано	Требуется установить расширение командой: <pre>create extension securityprofile;</pre> Подробно действия описаны в п.6.1.2
2.	Table (%s) does not exist. Probably extension (%s) is not created yet	Таблица (<i>имя таблицы</i>) не существует. Вероятно, расширение (<i>имя профиля</i>) еще не создано	Требуется установить расширение командой: <pre>create extension securityprofile;</pre> Подробно действия описаны в п.6.1.2
3.	Extension "securityprofile" has to be loaded using shared_preload_libraries	Расширение "securityprofile" должно быть загружено с помощью shared_preload_libraries	Для устранения ошибки в файле «postgresql.conf» прописать следующую строку: <pre>shared_preload_libraries = 'securityprofile'</pre> Подробно действия описаны в п.6.1.2
4.	You must be superuser to run this function	Вы должны быть суперпользователем, чтобы запустить эту функцию	Ошибка возникает при попытке с недостаточными привилегиями пользователя запустить функции управления расширениями, в том числе «securityprofile». Пользователю потребуются назначить дополнительные привилегии, в том числе на схему выполнив команды: <pre>GRANT CREATE ON SCHEMA securityprofile TO <имя пользователя>; GRANT USAGE ON SCHEMA securityprofile TO <имя пользователя>;</pre>
5.	Failed login attempts limit exceeded	Превышен лимит неудачных попыток входа в систему	Пользователю следует подождать установленное

№	Текст ошибки	Перевод	Пути исправления
			время или обратиться к администратору
6.	Password may not be reused. Try another	Пароль не может быть использован повторно. Попробовать другой	Сообщение возникает при нарушении парольной политики повторного использования пароля
7.	Value of string's length should be between min_border and max_border	Значение длины строки должно быть между min_border и max_border	<p>Ошибка возникает, при:</p> <ul style="list-style-type: none"> • смене пароля пользователя; • создании или переименовании профиля и задании строки спецсимволов. <p>Длина введенной строки не удовлетворяет предустановленным границам расширения:</p> <ul style="list-style-type: none"> • имени от 1 до 32; • пароля от 6 до 256; • строки спецсимволов от 0 до 32
8.	New password must have at least (%d) changes. profile-> password_min_changes_count	Новый пароль должен иметь не менее (количество символов) изменений. profile -> password_min_changes_count	Сообщение возникает при нарушении политики количества изменений в пароле
9.	User's password may not be changed right now. Password's minimim life time not expired yet	Пароль пользователя не может быть изменен прямо сейчас. Минимальное время жизни пароля еще не истекло	<p>В случае, когда инициализирован параметр «securityprofile.password_min_1ife_time» и пользователю присвоен пароль начинается период в течение которого пароль должен использоваться и не может быть изменен.</p> <p>Соотношение временных параметров пароля подробно описано в п. 6.1.3.9.2, а параметры парольных политик приведены в таблице</p> <p>Отключение пользовательского профиля парольной политики</p> <p>Чтобы отключить созданный пользовательский профиль (см. п. 6.1.3.3) и переключиться на профиль по умолчанию</p>

№	Текст ошибки	Перевод	Пути исправления
			<p>необходимо удалить (или закомментировать) параметр securityprofile.default_profile в конфигурационном файле postgresql.conf.</p> <p>При переключении на профиль по умолчанию требования к парольной политике существующих пользователей не изменяются, используются только для новых пользователей СУБД.</p> <p>18.5.1.1 Смена профиля парольной политики для пользователя</p> <p>Профиль можно сменить пользователю с помощью SQL-команды следующего синтаксиса:</p> <pre>SELECT securityprofile.bind_profile(['user'], ['profile_name']);</pre> <p>Пример:</p> <pre>SELECT securityprofile.bind_profile('manager', 'profile_hard');</pre> <p>.</p> <p>В данном случае необходимо дождаться окончания минимального времени действия пароля, либо</p>

№	Текст ошибки	Перевод	Пути исправления
			отключить параметр «securityprofile.password_min_life_time»
10.	Password is too short	Пароль слишком короткий	Сообщение возникает при попытке установить пароль, который не соответствует значению минимального количества символов (securityprofile.minimum_length)
11.	Password is too long	Пароль слишком длинный	Сообщение возникает при попытке установить пароль количество символов в котором превышает установленный параметр максимальной длины пароля (securityprofile.maximum_length)
12.	Password must not contain user name	Пароль не должен содержать имя пользователя	Сообщение возникает при проверке содержания пароля. В теле пароля не должно быть указания имени пользователя
13.	Password contains invalid characters	Пароль содержит недопустимые символы	Сообщение возникает, когда в пароле присутствуют специальные символы, не содержащиеся в справочнике специальных символов, заданных параметром «securityprofile.special_chars»
14.	Extension (%)s need existent role to check it's new password. Call CREATE ROLE without PASSWORD part then use ALTER ROLE ... PASSWORD to set new password properly	Расширению требуется уже существующая роль, для проверки ее нового пароля. Выполните запрос CREATE ROLE без установки пароля. Затем выполните запрос ALTER ROLE для установки пароля	Ошибка возникает при создании пользователя с указанием пароля. Подробно создание пользователя в п.6.2.4
15.	Can not check password validity	Невозможно проверить валидность пароля	Ошибка связана с особенностью реализации

№	Текст ошибки	Перевод	Пути исправления
	Password may be changed only by ALTER ROLE query	Пароль может быть изменен только запросом ALTER ROLE	парольных политик. Изменить пароль пользователя можно через команду ALTER ROLE
16.	User must change password. Password grace time expired	Пользователь должен изменить пароль. Срок действия льготного пароля истек	Ошибка возникает при истечении срока действия пароля
17.	User should change password. Password life time expired. Password still in use until grace time expired, so user may login on next try	Пользователь должен сменить пароль. Срок службы пароля истек. Пароль все еще используется до истечения льготного времени, поэтому пользователь может войти в систему при следующей попытке	Сообщение возникает при истечении срока действия пароля пользователя, если установлен параметр «securityprofile.password_grace_time» - время в секундах, в течение которого пользователь может использовать текущий пароль с напоминанием о необходимости его сменить до блокировки аккаунта. Время прибавляется к времени, установленному в securityprofile.password_life_time. Подробно взаимодействие параметров описано в пп. 6.1.3.9.1
18.	Account is locked forever	Учетная запись заблокирована навсегда	Учетную запись пользователя может разблокировать только Superuser. Ошибка возникает, если время блокировки установлено параметром «unlimited» (бесконечно)
19.	Account is locked till. Try later unlock_date	Учетная запись пользователя заблокирована до «даты». Попробуйте позже «даты»	Учетную запись пользователя может разблокировать только Superuser. Ошибка возникает, если время блокировки установлено на определенный период времени
20.	Extension "EXTENSION_NAME" need to be initialized by superuser	Расширение «securityprofile» должно быть инициализировано суперпользователем	Сообщение возникает при ошибке инициализации «securityprofile». Действия по

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

№	Текст ошибки	Перевод	Пути исправления
	Login as superuser and call "select securityprofile.synchronize()	Войдите в систему, как суперпользователь и вызовите команду select securityprofile. synchronize ()	исправлению описаны в подразделе 18.3
21.	Value should be between min_border and max_border	Значение должно быть между min_border и max_border	Ошибка возникает при установке значений профиля. В данном случае устанавливаемые параметры выходят за предустановленные границы
22.	Password minimum length can not be larger than maximum length, default_profile.password_max_len	Минимальная длина пароля не может быть больше максимальной длины, default_profile.password_max_len	Ошибка возникает при создании профиля для групп пользователей «securityprofile», если ошибочно были указаны противоречивые параметры, где минимальная длина пароля более максимальной длины
23.	Password maximum length can not be lesser than minimum length, default_profile.password_min_len	Максимальная длина пароля не может быть меньше минимальной длины, default_profile.password_min_len	Ошибка возникает при создании профиля для групп пользователей «securityprofile», если ошибочно были указаны противоречивые параметры, где максимальная длина пароля менее минимальной длины
24.	Default profile already exist. Specified name used by default profile	Профиль по умолчанию уже существует. Указанное имя используется профиль по умолчанию	Ошибка возникает при создании одноименного профиля существующему профилю securityprofile (default)
25.	Profile already exist	Профиль (имя профиля) уже существует	Ошибка может возникать при указании имени нового профиля, совпадающего с уже существующим профилем securityprofile
26.	Default profile can not be removed	Профиль по умолчанию не может быть удален	Поскольку securityprofile (default) является базовым

№	Текст ошибки	Перевод	Пути исправления
			профилем, то его нельзя удалить
27.	Profile have binded users	Профиль (имя профиля) имеет привязанных пользователей	Сообщение появляется при попытке удалить профайл с привязанными к нему пользователями. Для исправления ошибки необходимо перепривязать пользователей, выполнив команду select (имя схемы) bind_profile ('имя_профиля', 'имя_пользователя'). Команды управления профилями приведены в таблице 6.11
28.	Default profile can not be changed with function call	Профиль по умолчанию не может быть изменен с помощью вызова функции	Профиль по умолчанию не может быть изменен стандартными функциями. Изменения допустимы через редактирование конфигурационного файла postgresql.conf
29.	Profile name already in use, new_profile_name	Имя профиля (имя профиля) уже используется	Используйте другое имя создаваемого профиля
30.	Password minimum length can not be larger than maximum length, profile->password_max_len	Минимальная длина пароля не может быть больше максимальной длины, profile ->password_max_len)	Сообщение об ошибке возникает при формировании профиля для групп пользователей, при установлении длины пароля. Следует присвоить минимальной длине пароля (password_min_len) значение, которое не будет превышать ранее заданную максимальную длину пароля (password_max_len)
31.	Password maximum length can not be lesser than minimum length, profile->password_min_len	Максимальная длина пароля не может быть меньше минимальной длины, profile ->password_min_len	Сообщение об ошибке возникает при формировании профиля для групп пользователей, при

№	Текст ошибки	Перевод	Пути исправления
			установлении максимальной длины пароля. Следует присвоить максимальное значение длины пароля (password_max_len), превышающее ранее заданное значение минимальной длины пароля (password_min_len)
32.	Shared buffer hash table corrupted(profiles)	Общая буферная хэш-таблица повреждена (<i>имя профиля</i>)	Ошибки связаны с системным сбоем. Для их устранения рекомендуется: <ul style="list-style-type: none">• провести тестирование оперативной памяти;• перезагрузить СУБД;• восстановить БД из резервной копии
33.	Password encryption failed	Ошибка шифрования пароля	
34.	Cache entry does not exist	Запись в кэше не существует	
35.	Cache synchronization failed for background worker of securityprofile. Extension not installed	Фоновый процесс не может выполнить синхронизацию, расширение не установлено	Указать в файле PostgreSQL.conf параметр securityprofile.db_name = 'dbname' и/или установить расширение securityprofile в БД dbname

18.6. Ошибка авторизации

Ошибка авторизации в psql после установки СУБД «Jatoba» с настройками СУБД: «Язык и регион: English_USA» и «Кодировка: WIN1252» на английскую версию ОС Windows Server с выбранными при установке параметрами: «Time and currency format: Russian (Russia)» и «Keyboard or input method: Russian».

- 1) Запустить cmd.exe.
- 2) Выполнить:

```
psql -U postgres;
```

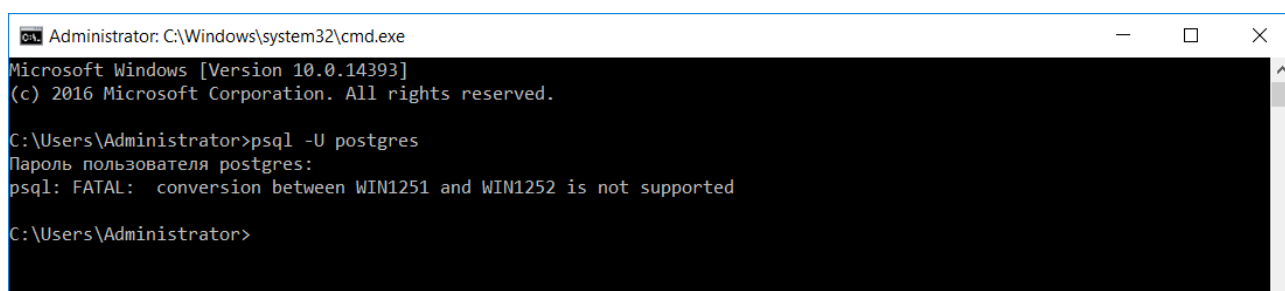
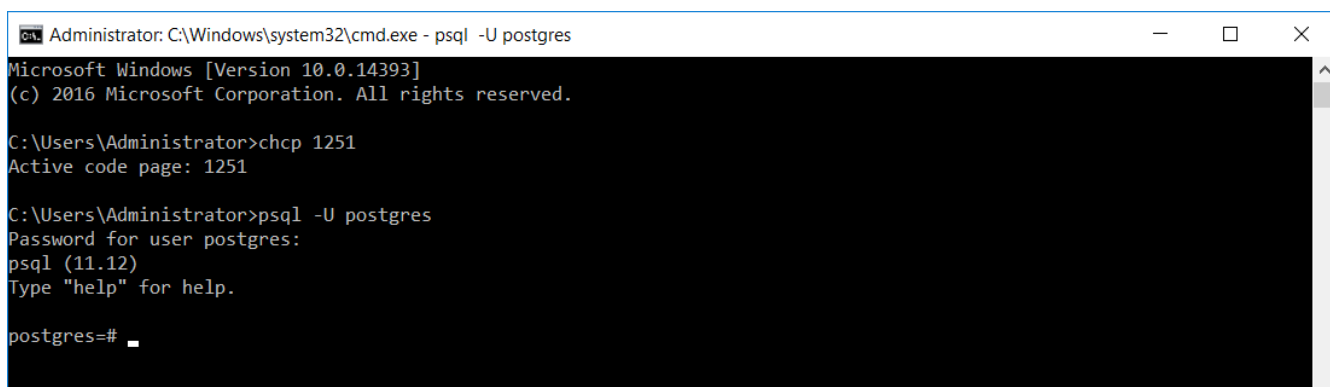


Рисунок 18.3 – Окно командной строки с ошибкой

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

3) Решение проблемы:

- перейти в панель управления (control panel);
- затем перейти в «Clock, Language, and Region/Language»;
- нажать «Advanced settings»;
- в секции «Override for Windows display language» выбрать «English (United Stated)»;
- нажать «Save»;
- снова запустить cmd.exe;
- ввести: chcp 1251;
- повторить вход в psql.



```
Administrator: C:\Windows\system32\cmd.exe - psql -U postgres
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>chcp 1251
Active code page: 1251

C:\Users\Administrator>psql -U postgres
Password for user postgres:
psql (11.12)
Type "help" for help.

postgres=#
```

Рисунок 18.4 – Окно командной строки без ошибок

18.7. Компонент «ja_seceventlog». Ошибка загрузки библиотеки

Сообщение об ошибке загрузки библиотеки

```
ja_seceventlog must be loaded via shared_preload_libraries
```

возникает, при применении установленных параметров в конфигурационных файлах SQL-командой:

```
SELECT pg_reload_conf ();
```

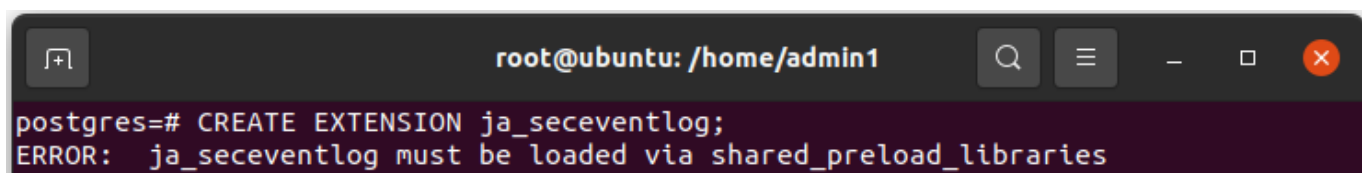


Рисунок 18.5 – Ошибка загрузки библиотеки

Для устранения возникшей ошибки, следует перезагрузить СУБД:

- в ОС Windows:

```
net stop JatobaServer
net start JatobaServer
```

- в GNU Linux:

```
systemctl restart jatoba-<ver>
```

18.8. Контактные данные службы технической поддержки

При невозможности самостоятельно решить возникшие трудности с СУБД «Jatoba» следует обратиться в службу технической поддержки ООО «Газинформсервис».

Таблица 18.2 – Контактные данные службы технической поддержки

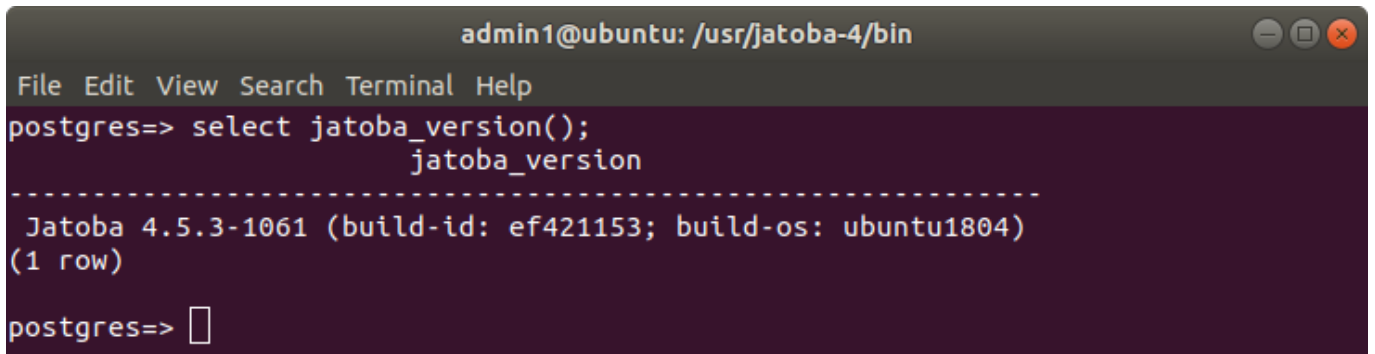
Телефон	8 (800) 700-09-87
Веб-сайт	https://www.gaz-is.ru/poddergka/zajavka.html#produkty
E-mail	support@gaz-is.ru

18.8.1. Версия изделия

Для скорейшего решения вопроса, рекомендуется сообщить в службу технической поддержки точную версию установленного экземпляра изделия.

Версию изделия можно узнать, выполнив команду:

```
SELECT jatoba_version();
```

A screenshot of a terminal window titled 'admin1@ubuntu: /usr/jatoba-4/bin'. The terminal has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The prompt is 'postgres=>'. The user has entered the command 'select jatoba_version();'. The output shows a single row with the value 'Jatoba 4.5.3-1061 (build-id: ef421153; build-os: ubuntu1804)'. The prompt is now 'postgres=>' with a cursor.

```
admin1@ubuntu: /usr/jatoba-4/bin
File Edit View Search Terminal Help
postgres=> select jatoba_version();
               jatoba_version
-----
Jatoba 4.5.3-1061 (build-id: ef421153; build-os: ubuntu1804)
(1 row)
postgres=> 
```

Рисунок 18.6 – Команда запроса версии изделия

ПРИЛОЖЕНИЕ 1

(обязательное)

Значение полей из файла pg_hba.conf

Значение поля «TYPE» представлены в таблице П.П.11.

Таблица П.1.1.1 – Значение поля «TYPE»

Значение поля	Описание
local	Сопоставляет попытки подключения с использованием Unix-сокетов. Без данной записи все соединения через Unix-сокеты будут запрещены. Данное поле не работает в ОС Windows Server
host ¹⁾	Соответствует попыткам подключения, выполненным с использованием TCP/IP. Записи хоста соответствуют попыткам подключения SSL или без SSL
hostssl	Соответствует попыткам подключения, выполненным с использованием TCP/IP и шифрованием SSL
hostnssl	Соответствует попыткам подключения, выполненным с использованием TCP/IP (без шифрования ssl)
¹⁾ Удаленное соединение TCP/IP будут невозможны, если сервер не запущен с подходящим значением для параметра конфигурации listen_addresses, поскольку по умолчанию выполняется прослушивание соединений TCP/IP только на локальном кольцевом адресе localhost	

Поле «DATABASE» указывает, какие имена баз данных соответствует данной строки. Значение поля «DATABASE» представлены в таблице П.П.12.

Таблица П.1.1.2 – Значение поля «DATABASE»

Значение поля	Описание
all	Указывает, что данная строка относится ко всем базам данных
sameuser	Указывает, что данная строка соответствует тому, что запрашиваемая база имеет то же имя, что и запрашиваемый пользователь
samerole	Указывает, что данная строка соответствует тому, что запрашиваемый пользователь должен быть членом роли с тем же именем, что и запрошенная база данных. Суперпользователи не считаются членами роли для целей samerole, если они не являются явными членами роли, прямо или косвенно
replication	Значение replication указывает, что запрашивается подключение репликации (в этом случае конкретная база данных не указывается)

Поле «USER» указывает, что данная строка соответствует конкретному имени пользователя. Значение all указывает, что оно соответствует всем пользователям. Можно указать несколько имен пользователей, разделяя их запятыми.

Поле «ADDRESS» указывает, что данная строка соответствует адресу клиентской машины. Данное поле может содержать имя хоста или диапазон IP-адресов, или одно из нижеупомянутых ключевых слов. Диапазон IP-адресов указывается с использованием

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

стандартных числовых обозначений для начального адреса диапазона, затем косой черты (/) и длины маски CIDR. Длина маски указывает количество старших битов IP-адреса клиента, которые должны совпадать. Биты справа от этого должны быть равны нулю в данном IP-адресе. Между IP-адресом, / и длиной маски CIDR не должно быть пробелов. Запись в формате IPv4 будет соответствовать только соединениям IPv4, а запись в формате IPv6 будет соответствовать только соединениям IPv6, даже если представленный адрес находится в диапазоне IPv4-in-IPv6.

Поле «IP-адрес IP-маска» используется в качестве альтернативы IP-адрес / длина маски. Вместо указания длины маски фактическая маска указывается в отдельном столбце. Например, 255.0.0.0 представляет длину маски CIDR IPv4 8, а 255.255.255.255 представляет длину маски CIDR 32.

Поле «METHOD» определяет, что данная строка будет осуществлять соединение по этому методу аутентификации. Значение поля «METHOD» представлены в таблице П.1.3.

Таблица П.1.1.3 – Значение поля «METHOD»

Значение поля	Описание
md5	Аутентификация осуществляется по паролю. По каналу связи передается пароль в виде хеша MD5.
password	Аутентификация осуществляется по паролю. Пароль передается в открытом виде.

ПРИЛОЖЕНИЕ 2

Перечень событий СУБД с распределением по категориям безопасности

Таблица П.2.1 – Перечень событий СУБД с распределением по категориям безопасности

Категория события/ Класс события	Код события	Название условия
Class 00 - Successful Completion		
Управление данными	00000	ERRCODE SUCCESSFUL COMPLETION
Class 01 - Warning		
Управление данными	01000	ERRCODE WARNING
Управление данными	0100C	ERRCODE WARNING DYNAMIC RESULT SETS RETURNED
Управление данными	01008	ERRCODE WARNING IMPLICIT ZERO BIT PADDING
Управление данными	01003	ERRCODE WARNING NULL VALUE ELIMINATED IN SET FUNCTION
Управление доступом	01007	ERRCODE WARNING PRIVILEGE NOT GRANTED
Управление доступом	01006	ERRCODE WARNING PRIVILEGE NOT REVOKED
Управление данными	01004	ERRCODE WARNING STRING DATA RIGHT TRUNCATION
Управление данными	01P01	ERRCODE WARNING DEPRECATED FEATURE
Class 02 - No Data		
Управление данными	02000	ERRCODE NO DATA
Управление данными	02001	ERRCODE NO ADDITIONAL DYNAMIC RESULT SETS RETURNED
Class 03 - SQL Statement Not Yet Complete		
Управление данными	03000	ERRCODE SQL STATEMENT NOT YET COMPLETE
Class 08 - Connection Exception		
Управление доступом	08000	ERRCODE CONNECTION EXCEPTION
Управление доступом	08003	ERRCODE CONNECTION DOES NOT EXIST
Управление доступом	08006	ERRCODE CONNECTION FAILURE
Управление доступом	08001	ERRCODE SQLCLIENT UNABLE TO ESTABLISH SQLCONNECTION
Управление доступом	08004	ERRCODE SQLSERVER REJECTED ESTABLISHMENT OF SQLCONNECTION
Управление доступом	08007	ERRCODE TRANSACTION RESOLUTION UNKNOWN
Управление доступом	08P01	ERRCODE PROTOCOL VIOLATION
Class 09 - Triggered Action Exception		
Управление данными	09000	ERRCODE TRIGGERED ACTION EXCEPTION
Class 0A - Feature Not Supported		
Управление данными	0A000	ERRCODE FEATURE NOT SUPPORTED
Class 0B - Invalid Transaction Initiation		
Управление данными	0B000	ERRCODE INVALID TRANSACTION INITIATION
Class 0F - Locator Exception		
Прочее	0F000	ERRCODE LOCATOR EXCEPTION
Прочее	0F001	ERRCODE L E INVALID SPECIFICATION
Class 0L - Invalid Grantor		
Управление доступом	0L000	ERRCODE INVALID GRANTOR
Управление доступом	0LP01	ERRCODE INVALID GRANT OPERATION
Class 0P - Invalid Role Specification		
Управление доступом	0P000	ERRCODE INVALID ROLE SPECIFICATION
Class 0Z - Diagnostics Exception		

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

Категория события/ Класс события	Код события	Название условия
Управление данными	0Z000	ERRCODE DIAGNOSTICS EXCEPTION
Управление данными	0Z002	ERRCODE STACKED DIAGNOSTICS ACCESSED WITHOUT ACTIVE HANDLER
Class 20 - Case Not Found		
Управление данными	20000	ERRCODE CASE NOT FOUND
Class 21 - Cardinality Violation		
Прочее	21000	ERRCODE CARDINALITY VIOLATION
Class 22 - Data Exception		
Управление данными	22000	ERRCODE DATA EXCEPTION
Управление данными	2202E	ERRCODE ARRAY SUBSCRIPT ERROR
Управление данными	22021	ERRCODE CHARACTER NOT IN REPERTOIRE
Управление данными	22008	ERRCODE DATETIME FIELD OVERFLOW
Контроль целостности	22008	ERRCODE DATETIME VALUE OUT OF RANGE
Управление данными	2202	ERRCODE ARRAY ELEMENT ERROR
Управление данными	22012	ERRCODE DIVISION BY ZERO
Управление данными	22005	ERRCODE ERROR IN ASSIGNMENT
Управление данными	2200B	ERRCODE ESCAPE CHARACTER CONFLICT
Управление данными	22022	ERRCODE INDICATOR OVERFLOW
Управление данными	22015	ERRCODE INTERVAL FIELD OVERFLOW
Управление данными	2201E	ERRCODE INVALID ARGUMENT FOR LOG
Управление данными	22013	ERRCODE INVALID PRECEDING OR FOLLOWING SIZE
Управление данными	22014	ERRCODE INVALID ARGUMENT FOR NTILE
Управление данными	22016	ERRCODE INVALID ARGUMENT FOR NTH VALUE
Управление данными	2201F	ERRCODE INVALID ARGUMENT FOR POWER FUNCTION
Управление данными	2201G	ERRCODE INVALID ARGUMENT FOR WIDTH BUCKET FUNCTION
Управление данными	22018	ERRCODE INVALID CHARACTER VALUE FOR CAST
Управление данными	22007	ERRCODE INVALID DATETIME FORMAT
Управление данными	22019	ERRCODE INVALID ESCAPE CHARACTER
Управление данными	2200D	ERRCODE INVALID ESCAPE OCTET
Управление данными	22025	ERRCODE INVALID ESCAPE SEQUENCE
Управление данными	22P06	ERRCODE NONSTANDARD USE OF ESCAPE CHARACTER
Управление данными	22010	ERRCODE INVALID INDICATOR PARAMETER VALUE
Управление данными	22023	ERRCODE INVALID PARAMETER VALUE
Управление данными	2201B	ERRCODE INVALID REGULAR EXPRESSION
Управление данными	2201W	ERRCODE INVALID ROW COUNT IN LIMIT CLAUSE
Управление данными	2201X	ERRCODE INVALID ROW COUNT IN RESULT OFFSET CLAUSE
Управление данными	2202H	ERRCODE INVALID TABLESAMPLE ARGUMENT
Управление данными	2202G	ERRCODE INVALID TABLESAMPLE REPEAT
Управление данными	22009	ERRCODE INVALID TIME ZONE DISPLACEMENT VALUE
Управление данными	2200C	ERRCODE INVALID USE OF ESCAPE CHARACTER
Управление данными	2200G	ERRCODE MOST SPECIFIC TYPE MISMATCH
Управление данными	22004	ERRCODE NULL VALUE NOT ALLOWED
Управление данными	22002	ERRCODE NULL VALUE NO INDICATOR PARAMETER
Управление данными	22003	ERRCODE NUMERIC VALUE OUT OF RANGE
Управление данными	22026	ERRCODE STRING DATA LENGTH MISMATCH
Управление данными	22001	ERRCODE STRING DATA RIGHT TRUNCATION
Управление данными	22011	ERRCODE SUBSTRING ERROR
Управление данными	22027	ERRCODE TRIM ERROR
Управление данными	22024	ERRCODE UNTERMINATED C STRING
Управление данными	2200F	ERRCODE ZERO LENGTH CHARACTER STRING

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

Категория события/ Класс события	Код события	Название условия
Управление данными	22P01	ERRCODE FLOATING POINT EXCEPTION
Управление данными	22P02	ERRCODE INVALID TEXT REPRESENTATION
Управление данными	22P03	ERRCODE INVALID BINARY REPRESENTATION
Управление данными	22P04	ERRCODE BAD COPY FILE FORMAT
Управление данными	22P05	ERRCODE UNTRANSLATABLE CHARACTER
Управление данными	2200L	ERRCODE NOT AN XML DOCUMENT
Управление данными	2200M	ERRCODE INVALID XML DOCUMENT
Управление данными	2200N	ERRCODE INVALID XML CONTENT
Управление данными	2200S	ERRCODE INVALID XML COMMENT
Управление данными	2200T	ERRCODE INVALID XML PROCESSING INSTRUCTION
Управление данными	2200H	ERRCODE SEQUENCE GENERATOR LIMIT EXCEEDED
Управление данными	22030	ERRCODE DUPLICATE JSON OBJECT KEY VALUE
Управление данными	22031	ERRCODE INVALID ARGUMENT FOR SQL JSON DATETIME FUNCTION
Управление данными	22032	ERRCODE INVALID JSON TEXT
Управление данными	22033	ERRCODE INVALID SQL JSON SUBSCRIPT
Управление данными	22034	ERRCODE MORE THAN ONE SQL JSON ITEM
Управление данными	22035	ERRCODE NO SQL JSON ITEM
Управление данными	22036	ERRCODE NON NUMERIC SQL JSON ITEM
Управление данными	22037	ERRCODE NON UNIQUE KEYS IN A JSON OBJECT
Управление данными	22038	ERRCODE SINGLETON SQL JSON ITEM REQUIRED
Управление данными	22039	ERRCODE SQL JSON ARRAY NOT FOUND
Управление данными	2203A	ERRCODE SQL JSON MEMBER NOT FOUND
Управление данными	2203B	ERRCODE SQL JSON NUMBER NOT FOUND
Управление данными	2203C	ERRCODE SQL JSON OBJECT NOT FOUND
Управление данными	2203D	ERRCODE TOO MANY JSON ARRAY ELEMENTS
Управление данными	2203E	ERRCODE TOO MANY JSON OBJECT MEMBERS
Управление данными	2203F	ERRCODE SQL JSON SCALAR REQUIRED
Class 23 - Integrity Constraint Violation		
Контроль целостности	23000	ERRCODE INTEGRITY CONSTRAINT VIOLATION
Контроль целостности	23001	ERRCODE RESTRICT VIOLATION
Контроль целостности	23502	ERRCODE NOT NULL VIOLATION
Контроль целостности	23503	ERRCODE FOREIGN KEY VIOLATION
Контроль целостности	23505	ERRCODE UNIQUE VIOLATION
Контроль целостности	23514	ERRCODE CHECK VIOLATION
Контроль целостности	23P01	ERRCODE EXCLUSION VIOLATION
Class 24 - Invalid Cursor State		
Прочее	24000	ERRCODE INVALID CURSOR STATE
Class 25 - Invalid Transaction State		
Управление данными	25000	ERRCODE INVALID TRANSACTION STATE
Управление данными	25001	ERRCODE ACTIVE SQL TRANSACTION
Управление данными	25002	ERRCODE BRANCH TRANSACTION ALREADY ACTIVE
Управление данными	25008	ERRCODE HELD CURSOR REQUIRES SAME ISOLATION LEVEL
Управление данными	25003	ERRCODE INAPPROPRIATE ACCESS MODE FOR BRANCH TRANSACTION
Управление данными	25004	ERRCODE INAPPROPRIATE ISOLATION LEVEL FOR BRANCH TRANSACTION
Управление данными	25005	ERRCODE NO ACTIVE SQL TRANSACTION FOR BRANCH TRANSACTION
Управление данными	25006	ERRCODE READ ONLY SQL TRANSACTION

Категория события/ Класс события	Код события	Название условия
Управление данными	25007	ERRCODE SCHEMA AND DATA STATEMENT MIXING NOT SUPPORTED
Управление данными	25P01	ERRCODE NO ACTIVE SQL TRANSACTION
Управление данными	25P02	ERRCODE IN FAILED SQL TRANSACTION
Управление данными	25P03	ERRCODE IDLE IN TRANSACTION SESSION TIMEOUT
Class 26 - Invalid SQL Statement Name		
Управление данными	26000	ERRCODE INVALID SQL STATEMENT NAME
Class 27 - Triggered Data Change Violation		
Управление доступом	27000	ERRCODE TRIGGERED DATA CHANGE VIOLATION
Class 28 - Invalid Authorization Specification		
Идентификация	28000	ERRCODE INVALID AUTHORIZATION SPECIFICATION
Идентификация	28P01	ERRCODE INVALID PASSWORD
Class 2B - Dependent Privilege Descriptors Still Exist		
Управление доступом	2B000	ERRCODE DEPENDENT PRIVILEGE DESCRIPTORS STILL EXIST
Управление доступом	2BP01	ERRCODE DEPENDENT OBJECTS STILL EXIST
Class 2D - Invalid Transaction Termination		
Управление данными	2D000	ERRCODE INVALID TRANSACTION TERMINATION
Class 2F - SQL Routine Exception		
Управление доступом	2F000	ERRCODE SQL ROUTINE EXCEPTION
Управление доступом	2F005	ERRCODE S R E FUNCTION EXECUTED NO RETURN STATEMENT
Управление доступом	2F002	ERRCODE S R E MODIFYING SQL DATA NOT PERMITTED
Управление доступом	2F003	ERRCODE S R E PROHIBITED SQL STATEMENT ATTEMPTED
Управление доступом	2F004	ERRCODE S R E READING SQL DATA NOT PERMITTED
Class 34 - Invalid Cursor Name		
Прочее	34000	ERRCODE INVALID CURSOR NAME
Class 38 - External Routine Exception		
Управление доступом	38000	ERRCODE EXTERNAL ROUTINE EXCEPTION
Управление доступом	38001	ERRCODE E R E CONTAINING SQL NOT PERMITTED
Управление доступом	38002	ERRCODE E R E MODIFYING SQL DATA NOT PERMITTED
Управление доступом	38003	ERRCODE E R E PROHIBITED SQL STATEMENT ATTEMPTED
Управление доступом	38004	ERRCODE E R E READING SQL DATA NOT PERMITTED
Class 39 - External Routine Invocation Exception		
Управление данными	39000	ERRCODE EXTERNAL ROUTINE INVOCATION EXCEPTION
Управление данными	39001	ERRCODE E R I E INVALID SQLSTATE RETURNED
Управление данными	39004	ERRCODE E R I E NULL VALUE NOT ALLOWED
Управление данными	39P01	ERRCODE E R I E TRIGGER PROTOCOL VIOLATED
Управление данными	39P02	ERRCODE E R I E SRF PROTOCOL VIOLATED
Управление данными	39P03	ERRCODE E R I E EVENT TRIGGER PROTOCOL VIOLATED
Class 3B - Savepoint Exception		
Резервное копирование	3B000	ERRCODE SAVEPOINT EXCEPTION
Резервное копирование	3B001	ERRCODE S E INVALID SPECIFICATION
Class 3D - Invalid Catalog Name		
Прочее	3D000	ERRCODE INVALID CATALOG NAME
Class 3F - Invalid Schema Name		
Управление данными	3F000	ERRCODE INVALID SCHEMA NAME
Class 40 - Transaction Rollback		
Управление данными	40000	ERRCODE TRANSACTION ROLLBACK

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

Категория события/ Класс события	Код события	Название условия
Управление данными	40002	ERRCODE T R INTEGRITY CONSTRAINT VIOLATION
Управление данными	40001	ERRCODE T R SERIALIZATION FAILURE
Управление данными	40003	ERRCODE T R STATEMENT COMPLETION UNKNOWN
Управление данными	40P01	ERRCODE T R DEADLOCK DETECTED
Class 42 - Syntax Error or Access Rule Violation		
Управление доступом	42000	ERRCODE SYNTAX ERROR OR ACCESS RULE VIOLATION
Управление данными	42601	ERRCODE SYNTAX ERROR
Управление доступом	42501	ERRCODE INSUFFICIENT PRIVILEGE
Управление доступом	42846	ERRCODE CANNOT COERCE
Управление доступом	42803	ERRCODE GROUPING ERROR
Управление данными	42P20	ERRCODE WINDOWING ERROR
Управление данными	42P19	ERRCODE INVALID RECURSION
Идентификация	42830	ERRCODE INVALID FOREIGN KEY
Управление данными	42602	ERRCODE INVALID NAME
Управление данными	42622	ERRCODE NAME TOO LONG
Управление данными	42939	ERRCODE RESERVED NAME
Контроль целостности данных	42804	ERRCODE DATATYPE MISMATCH
Контроль целостности данных	42P18	ERRCODE INDETERMINATE DATATYPE
Управление данными	42P21	ERRCODE COLLATION MISMATCH
Управление данными	42P22	ERRCODE INDETERMINATE COLLATION
Контроль целостности данных	42809	ERRCODE WRONG OBJECT TYPE
Управление данными	428C9	ERRCODE GENERATED ALWAYS
Управление данными	42703	ERRCODE UNDEFINED COLUMN
Управление данными	42883	ERRCODE UNDEFINED FUNCTION
Управление данными	42P01	ERRCODE UNDEFINED TABLE
Управление данными	42P02	ERRCODE UNDEFINED PARAMETER
Управление данными	42704	ERRCODE UNDEFINED OBJECT
Управление данными	42701	ERRCODE DUPLICATE COLUMN
Управление данными	42P03	ERRCODE DUPLICATE CURSOR
Управление данными	42P04	ERRCODE DUPLICATE DATABASE
Управление данными	42723	ERRCODE DUPLICATE FUNCTION
Управление данными	42P05	ERRCODE DUPLICATE PSTATEMENT
Управление данными	42P06	ERRCODE DUPLICATE SCHEMA
Управление данными	42P07	ERRCODE DUPLICATE TABLE
Управление данными	42712	ERRCODE DUPLICATE ALIAS
Управление данными	42710	ERRCODE DUPLICATE OBJECT
Управление данными	42702	ERRCODE AMBIGUOUS COLUMN
Управление данными	42725	ERRCODE AMBIGUOUS FUNCTION
Управление данными	42P08	ERRCODE AMBIGUOUS PARAMETER
Управление данными	42P09	ERRCODE AMBIGUOUS ALIAS
Управление данными	42P10	ERRCODE INVALID COLUMN REFERENCE
Управление данными	42611	ERRCODE INVALID COLUMN DEFINITION
Управление данными	42P11	ERRCODE INVALID CURSOR DEFINITION
Управление данными	42P12	ERRCODE INVALID DATABASE DEFINITION
Управление данными	42P13	ERRCODE INVALID FUNCTION DEFINITION
Управление данными	42P14	ERRCODE INVALID PSTATEMENT DEFINITION
Управление данными	42P15	ERRCODE INVALID SCHEMA DEFINITION
Управление данными	42P16	ERRCODE INVALID TABLE DEFINITION
Управление данными	42P17	ERRCODE INVALID OBJECT DEFINITION

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

Категория события/ Класс события	Код события	Название условия
Class 44 - WITH CHECK OPTION Violation		
Управление доступом	44000	ERRCODE WITH CHECK OPTION VIOLATION
Class 53 - Insufficient Resources		
Прочее	53000	ERRCODE INSUFFICIENT RESOURCES
Прочее	53100	ERRCODE DISK FULL
Прочее	53200	ERRCODE OUT OF MEMORY
Прочее	53300	ERRCODE TOO MANY CONNECTIONS
Прочее	53400	ERRCODE CONFIGURATION LIMIT EXCEEDED
Class 54 - Program Limit Exceeded		
Управление данными	54000	ERRCODE PROGRAM LIMIT EXCEEDED
Управление данными	54001	ERRCODE STATEMENT TOO COMPLEX
Управление данными	54011	ERRCODE TOO MANY COLUMNS
Управление данными	54023	ERRCODE TOO MANY ARGUMENTS
Class 55 - Object Not In Prerequisite State		
Управление данными	55000	ERRCODE OBJECT NOT IN PREREQUISITE STATE
Управление данными	55006	ERRCODE OBJECT IN USE
Управление данными	55P02	ERRCODE CANT CHANGE RUNTIME PARAM
Управление данными	55P03	ERRCODE LOCK NOT AVAILABLE
Управление данными	55P04	ERRCODE UNSAFE NEW ENUM VALUE USAGE
Class 57 - Operator Intervention		
Прочее	57000	ERRCODE OPERATOR INTERVENTION
Прочее	57014	ERRCODE QUERY CANCELED
Прочее	57P01	ERRCODE ADMIN SHUTDOWN
Прочее	57P02	ERRCODE CRASH SHUTDOWN
Прочее	57P03	ERRCODE CANNOT CONNECT NOW
Прочее	57P04	ERRCODE DATABASE DROPPED
Прочее	57P05	ERRCODE IDLE SESSION TIMEOUT
Class 58 - System Error		
Прочее	58000	ERRCODE SYSTEM ERROR
Прочее	58030	ERRCODE IO ERROR
Управление данными	58P01	ERRCODE UNDEFINED FILE
Управление данными	58P02	ERRCODE DUPLICATE FILE
Class 72 - Snapshot Failure		
Управление данными	72000	ERRCODE SNAPSHOT TOO OLD
Class F0 - Configuration File Error		
Прочее	F0000	ERRCODE CONFIG FILE ERROR
Прочее	F0001	ERRCODE LOCK FILE EXISTS
Class HV - Foreign Data Wrapper Error (SQL/MED)		
Управление данными	HV000	ERRCODE FDW ERROR
Управление данными	HV005	ERRCODE FDW COLUMN NAME NOT FOUND
Управление данными	HV002	ERRCODE FDW DYNAMIC PARAMETER VALUE NEEDED
Управление данными	HV010	ERRCODE FDW FUNCTION SEQUENCE ERROR
Управление данными	HV021	ERRCODE FDW INCONSISTENT DESCRIPTOR INFORMATION
Управление данными	HV024	ERRCODE FDW INVALID ATTRIBUTE VALUE
Управление данными	HV007	ERRCODE FDW INVALID COLUMN NAME
Управление данными	HV008	ERRCODE FDW INVALID COLUMN NUMBER
Управление данными	HV004	ERRCODE FDW INVALID DATA TYPE
Управление данными	HV006	ERRCODE FDW INVALID DATA TYPE DESCRIPTORS
Управление данными	HV091	ERRCODE FDW INVALID DESCRIPTOR FIELD IDENTIFIER
Управление данными	HV00B	ERRCODE FDW INVALID HANDLE
Управление данными	HV00C	ERRCODE FDW INVALID OPTION INDEX
№ изменения: _____		Подпись отв. лица: _____
		Дата внесения изм: _____

Категория события/ Класс события	Код события	Название условия
Управление данными	HV00D	ERRCODE FDW INVALID OPTION NAME
Управление данными	HV090	ERRCODE FDW INVALID STRING LENGTH OR BUFFER LENGTH
Управление данными	HV00A	ERRCODE FDW INVALID STRING FORMAT
Управление данными	HV009	ERRCODE FDW INVALID USE OF NULL POINTER
Управление данными	HV014	ERRCODE FDW TOO MANY HANDLES
Управление данными	HV001	ERRCODE FDW OUT OF MEMORY
Управление данными	HV00P	ERRCODE FDW NO SCHEMAS
Управление данными	HV00J	ERRCODE FDW OPTION NAME NOT FOUND
Управление данными	HV00K	ERRCODE FDW REPLY HANDLE
Управление данными	HV00Q	ERRCODE FDW SCHEMA NOT FOUND
Управление данными	HV00R	ERRCODE FDW TABLE NOT FOUND
Управление данными	HV00L	ERRCODE FDW UNABLE TO CREATE EXECUTION
Управление данными	HV00M	ERRCODE FDW UNABLE TO CREATE REPLY
Управление данными	HV00N	ERRCODE FDW UNABLE TO ESTABLISH CONNECTION
Class P0 - PL/pgSQL Error		
Управление данными	P0000	ERRCODE PLPGSQL ERROR
Управление данными	P0001	ERRCODE RAISE EXCEPTION
Управление данными	P0002	ERRCODE NO DATA FOUND
Управление данными	P0003	ERRCODE TOO MANY ROWS
Управление данными	P0004	ERRCODE ASSERT FAILURE
Class XX - Internal Error		
Прочее	XX000	ERRCODE INTERNAL ERROR
Управление данными	XX001	ERRCODE DATA CORRUPTED
Управление данными	XX002	ERRCODE INDEX CORRUPTED

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

SMTP (Simple Mail Transfer Protocol) – протокол, используемый для передачи электронной почты.

ZULIP – веб-сервис для обмена сообщениями и организации обсуждений с использованием технологии real-time.

Администратор СУБД – субъект доступа, выполняющий административные функции в СУБД и наделенный правами:

- создавать учетные записи пользователей СУБД;
- модифицировать, блокировать и удалять учетные записи пользователей СУБД;
- назначать права доступа пользователям СУБД к объектам доступа СУБД;
- управлять конфигурацией СУБД;
- создавать, подключать БД.

Администратор СУБД имеет атрибут SUPERUSER и/или обладает системной учетной записью «postgres».

Администратор БД – субъект доступа, выполняющий административные функции в БД и наделенный правами:

- создавать учетные записи пользователей БД;
- модифицировать, блокировать и удалять учетные записи пользователей БД;
- управлять конфигурацией БД;
- назначать права доступа пользователям БД (пользователей информационной системы) к объектам доступа БД;
- создавать резервные копии БД и восстанавливать БД из резервной копии;
- создавать, модифицировать и удалять процедуры (программный код), хранимые в БД.

Администратор БД имеет атрибут CREATEROLE, и возможные атрибуты BYPASSRLS, REPLICATION, а также прочие системные привилегии относительно БД, кроме атрибута CREATEDB.

Пользователь СУБД – субъект доступа с назначенным атрибутом LOGIN, наделенный правами:

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

- создавать и манипулировать объектами доступа БД (таблица, запись или столбец, поле, представление и иные объекты доступа);
- выполнять процедуры (программный код), хранимые в БД.

Пользователь БД – субъект доступа, имеющий доступ к ограниченному перечню БД и объектов БД. Имеющий следующий набор привилегий:

- создавать и манипулировать объектами доступа БД (таблица, запись или столбец, поле, представление и иные объекты доступа);
- выполнять процедуры (программный код), хранимые в БД.

Пользователь БД имеет обязательный атрибут LOGIN.

Безусловная блокировка пользователя – это ограничение пользователя в возможности устанавливать новую сессию с СУБД. Безусловная блокировка имеет приоритет над ограничениями, накладываемыми парольными политикам (блокировка вследствие истечения срока действия пароля, временные блокировки при исчерпании попыток ввода пароля и т.п.), применяется независимо от них и не зависит от применяемого метода аутентификации пользователей. Снятие безусловной блокировки не снимает блокировок по парольным политикам и наоборот.

Завершение сессии пользователя – принудительное завершение открытой сессии пользователя с БД/СУБД в заданном режиме.

KNN (K-nearest neighbors) – это алгоритм машинного обучения, используемый для решения задач классификации и регрессии. Алгоритм KNN основан на идее, что объекты, которые находятся рядом в пространстве признаков, вероятно, относятся к одной категории.

Строчные буквы – это маленькие буквы, которые используются в письме и не превышают размеры строки.

Прописные буквы или заглавные – это графические знаки большего размера, которые превышают границы строки.

Лексема — это нормализованный фрагмент текста, в котором разные словоформы приведены к одной. Лексемы используются для индексации и поиска документов.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

API	—	Application programming interface
CIDR	—	Classless Inter-Domain Routing
CIS	—	Center for Internet Security
CSV	—	Comma-Separated Values
DDL	—	Data Definition Language
DML	—	Data Manipulation Language
DNS	—	Domain Name System
HDD	—	Hard Disk Drive
HTTP	—	HyperText Transfer Protocol
HTTPS	—	HyperText Transfer Protocol Secure
IEC	—	International Electrotechnical Commission
IIS	—	Internet Information Services
IP	—	Internet Protocol
ISO	—	International Organization for Standardization
LDAP	—	Lightweight Directory Access Protocol
SQL	—	Structured Query Language
SSL	—	Secure Sockets Layer
SSPI	—	Security Support Provider Interface
TCP	—	Transmission Control Protocol
АРМ	—	Автоматизированное рабочее место
БД	—	База данных
ИАФ	—	Идентификация и аутентификация
ИС	—	Информационная система
ОДТ	—	Обеспечение доступности информации
ОЗУ	—	Оперативное запоминающее устройство

ООО	–	Общество с ограниченной ответственностью
ОС	–	Операционная система
ОЦЛ	–	Обеспечение целостности
РСБ	–	Регистрация событий безопасности
СВТ	–	Средство вычислительной техники
СУБД	–	Система управления базами данных
ТП	–	Табличное пространство
УЗ	–	Учетная запись
УПД	–	Управление доступом
ФБО	–	Функция безопасности объекта оценки
ФСТЭК России	–	Федеральная служба по техническому и экспортному контролю России
ЭВМ	–	Электронно-вычислительная машина

[illegible]

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

237
643.72410666.00067-07 95 01

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