

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ «ЈАТОВА»

Руководство по настройке. Часть 6.
Формирование отчетов производительности СУБД.
Компонент «pg_Profile»

643.72410666.00067-07 98 01-06

Листов 79

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

АННОТАЦИЯ

В документе приведены сведения, необходимые для настройки и использования компонента pg_Profile, который позволяет собирать и просматривать параметры и метрики системы управления базами данных «Jatoba» (далее – СУБД «Jatoba»). Настоящее руководство предназначено для администратора СУБД «Jatoba».

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



Все примеры в данном документе приведены для СУБД «Jatoba» версии ядра 5.x, для других версий все шаги выполняются аналогично, разница состоит в именах директорий.

Например, СУБД «Jatoba» версии 6.x по умолчанию устанавливается в директорию:

- ОС Windows – «C:\Program Files\GIS\Jatoba\6\bin»;
- ОС Linux – «/usr/jatoba-6/bin».

Для СУБД «Jatoba» версии ядра 4/5/6 используется версия компонента — 4.6



Важная информация

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

Степени важности примечаний, применяемые в документе:

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



Важная информация – указания, требующие особого внимания



Дополнительная информация – указания, позволяющие упростить работу с изделием

СОДЕРЖАНИЕ

1. Назначение компонента.....	7
1.1. Функциональные возможности	7
1.2. Условия применения.....	8
2. Установка.....	9
2.1. Установка компонента на ОС Windows.....	9
2.2. Установка компонента ОС GNU/Linux	10
2.3. Рекомендованные настройки postgresql.conf.....	11
2.4. Установка расширения pg_Profile.....	13
2.4.1. Установка расширений в БД «postgres».....	13
2.4.2. Установка расширений в служебную БД.....	14
2.4.3. Установка внешнего сервера для служебной БД.....	17
2.5. Дополнительные настройки «postgresql.conf».....	21
3. Использование	22
3.1. Управление соединениями с серверами	22
3.2. Сбор данных о размерах страниц	23
3.3. Снимки состояния БД	24
3.3.1. Снятие снапшотов	26
3.3.2. Время жизни снапшотов	27
3.3.3. Описание событий.....	27
3.4. Baseline	28
3.4.1. Функции управления baseline	28
3.5. Экспорт / импорт данных.....	30
3.5.1. Экспорт данных.....	30
3.5.2. Импорт данных.....	31
3.6. Создание отчетов	32
3.6.1. Обычные отчеты.....	32
3.6.2. Дифференциальные отчеты.....	33
4. Данные в отчетах	35
4.1. Server statistics (Статистика сервера).....	35
4.1.1. Database statistics (Статистика базы данных).....	35
4.1.2. Statement statistics by database	36
4.1.3. Statement statistics by database (Статистика запросов по базе данных).....	37
4.1.4. Cluster statistics (Статистика кластера)	38
4.1.5. WAL statistics (Статистика WAL)	40
4.1.6. Tablespace statistics (Статистика табличных пространств).....	41
4.2. SQL Query statistics (Статистика SQL-запросов).....	41
4.2.1. Top SQL by elapsed time (Топ SQL-запросов по затраченному времени).....	42

4.2.2. Top SQL by planning time (Топ SQL-запросов по времени планирования)	43
4.2.3. Top SQL by execution time (Топ SQL-запросов по времени выполнения)	43
4.2.4. Top SQL by executions (Топ SQL-запросов по количеству выполнений)	44
4.2.5. Top SQL by I/O wait time (Топ SQL-запросов по времени ожидания ввода/вывода)	45
4.2.6. Top SQL by shared blocks fetched (Топ SQL-запросов по выбранным общим блокам)	46
4.2.7. Top SQL by shared blocks read (Топ SQL-запросов по количеству прочитанных разделяемых блоков)	47
4.2.8. Top SQL by shared blocks dirtied (Топ SQL-запросов по заполненным разделяемым блокам)	48
4.2.9. Top SQL by shared blocks written (Топ SQL-запросов по записи общих блоков)	49
4.2.10. Top SQL by WAL size (Топ SQL-запросов по размеру WAL)	49
4.2.11. Complete list of SQL texts (Полный список текстов SQL)	50
4.2.12. Top SQL by temp usage (Топ SQL-запросов по использованию временных данных)	51
4.3. Rusage statistics (Статистика использования ресурсов)	52
4.3.1. Top SQL by system and user time (Топ SQL-запросов по потреблению системного и пользовательского времени)	52
4.3.2. Top SQL by reads/writes done by filesystem layer (Топ SQL-запросов по количеству операций чтения/записи, выполняемых на уровне файловой системы)	53
4.3.3. Top SQL by JIT elapsed time (Топ SQL-запросов отсортированных по общему JIT-времени)	54
4.3.4. Complete list of SQL texts (Полный список текстов SQL-запросов)	55
4.4. Schema object statistics (Статистика объекта схемы)	56
4.4.1. Top tables by estimated sequentially scanned volume (Топ таблиц по предполагаемому объему последовательного сканирования)	56
4.4.2. Top tables by blocks fetched (Топ таблиц по выбранным блокам)	57
4.4.3. Top tables by blocks read (Топ таблиц по прочитанным блокам)	57
4.4.4. Top DML tables (Топ таблиц по количеству операций DML)	58
4.4.5. Top tables by updated/deleted tuples (Топ таблиц по обновленным/удаленным записям)	59
4.4.6. Top growing tables (Топ таблиц по увеличению размера)	60
4.4.7. Top indexes by blocks fetched (Топ индексов по выбранным блокам)	60
4.4.8. Top indexes by blocks read (Топ индексов по прочитанным блокам)	61
4.4.9. Top growing indexes (Топ таблиц по увеличению объемов индексов)	61
4.4.10. Unused indexes (Неиспользуемые индексы)	62
4.5. User function statistics (Статистика функций пользователя)	63
4.5.1. Top functions by total time (Топ функций по общему времени)	63
4.5.2. Top functions by executions (Топ функций по исполнению)	63
4.5.3. Top trigger functions by total time (Топ триггерных функций по общему времени)	64
4.6. Vacuum-related stats (Статистика, связанная с вакуумом)	64
4.6.1. Top tables by vacuum operations (Топ таблиц по статистике вакуума)	65
4.6.2. Top tables by analyze operations (Топ таблиц по операциям анализа)	65
4.6.3. Top indexes by estimated vacuum load (Топ индексов по расчетной нагрузке вакуумного ввода/вывода)	66

4.6.4. Top tables by dead tuples ratio (Топ таблиц по проценту наличия удаленных записей)	66
4.6.5. Top tables by modified tuples ratio (Топ таблиц по проценту измененных записей)	67
4.7. Cluster settings during the report interval (Настройка кластера во время отчетного интервала).....	68
4.8. Отчеты по компоненту «ja_Hipe_Cluster» (Citrus)	69
4.8.1. Nodes	69
4.8.2. Connectivity between all nodes.....	69
4.8.3. Tables (citrus_tables)	69
4.8.4. Shards	70
4.8.5. Blocked queries (citrus_lock_waits).....	71
4.8.6. Query statistics (citrus_stat_statements)	71
4.8.7. Rebalance progress (get_rebalance_progress)	72
4.8.8. Configuration parameters.....	72
4.8.9. Active tenants (citrus_stat_tenants)	73
5. Дополнительная информация.....	74
Термины и определение	75
Перечень сокращений.....	78

1. НАЗНАЧЕНИЕ КОМПОНЕНТА

pg_Profile – компонент, позволяющий собирать и просматривать параметры и метрики функционирования БД в разное время, а также строить отчеты по этим данным и сравнивать их между собой для выявления проблемных мест.

Для сбора параметров и метрик БД используется система снимков.

Анализ снимков состояния позволяет выявить проблемные участки путем просмотра или сравнения данных из разных снимков.

Снимки можно снимать как с текущей БД, так и с любых других БД, к которым есть доступ.

pg_Profile допускает экспортировать и импортировать собранную статистику в свои служебные таблицы.

Компонент реализован в формате расширения для СУБД PostgreSQL/Jatoba, установка которого регламентирована соответствующими правилами СУБД.



Для установки расширения необходимо заранее установить другие дополнительные расширения, на работе которых основана работа pg_profile, а именно – plpgsql и dblink.

Для получения данных в разделе отчета rusage_statistics необходимо установить расширение pg_stat_kcache и включить его параметр pg_stat_kcache.track_planning = 'on' в postgresql.conf. Если эти данные не нужны, то расширение можно не устанавливать.

1.1. Функциональные возможности

Компонент автоматизации работы администратора СУБД обладает следующими функциональными возможностями:

- 1) выявление и анализ наиболее ресурсоемких операций и SQL-запросов;
- 2) гибкое формирование итоговых отчетов по накопленной статистической информации.

Компонент предоставляет для просмотра информацию о SQL-запросах, выполнение которых по разным характеристикам занимало больше всего времени. Количество таких

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

запросов в формируемых отчетах определяется соответствующим параметром компонента, описанным в разделе 2.5.

Компонент обеспечивает сбор и хранение одномоментной и ретроспективной дискретной статистической информации о работе отдельных компонентов СУБД при выполнении SQL-запросов. Снятие статистической информации само по себе является затратной операцией, которая может вызывать блокировку отдельных объектов СУБД и влиять на производительность работы пользователей. В компоненте предусмотрен соответствующий механизм, основанный на использовании параметра СУБД `lock_timeout`, прерывающий построение снимка, который занимает более 3 секунд времени.

1.2. Условия применения

Компонент `pg_Profile` может использоваться совместно с СУБД «Jatoba» версий 1.x – 5.x.

2. УСТАНОВКА

2.1. Установка компонента на ОС Windows

Компонент устанавливается в составе СУБД «Jatoba» под управлением ОС Windows при первичной установке.

а) в окне «Выбор типа установки» следует выбрать тип установки «Выборочная» (см. рис. 2.1);

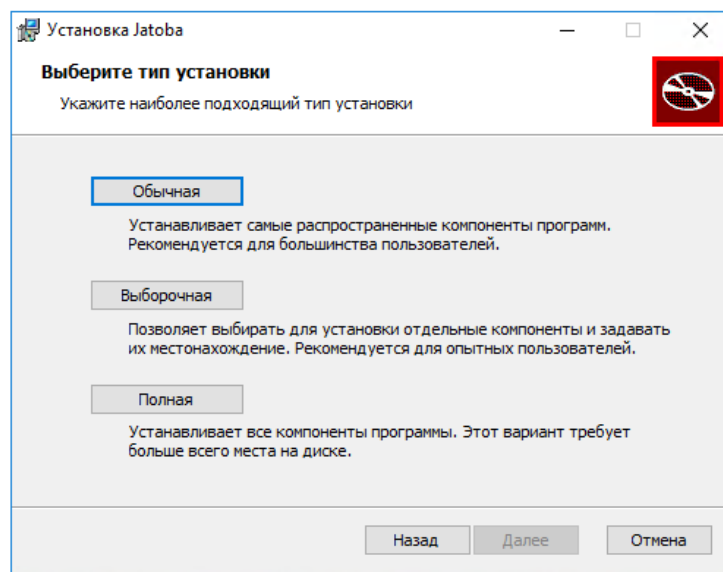


Рисунок 2.1 – Окно выбора типа установки

б) в окне «Выборочная установка» выбрать «Профилирование запросов (pg_profile)» (см. рис. 2.2);

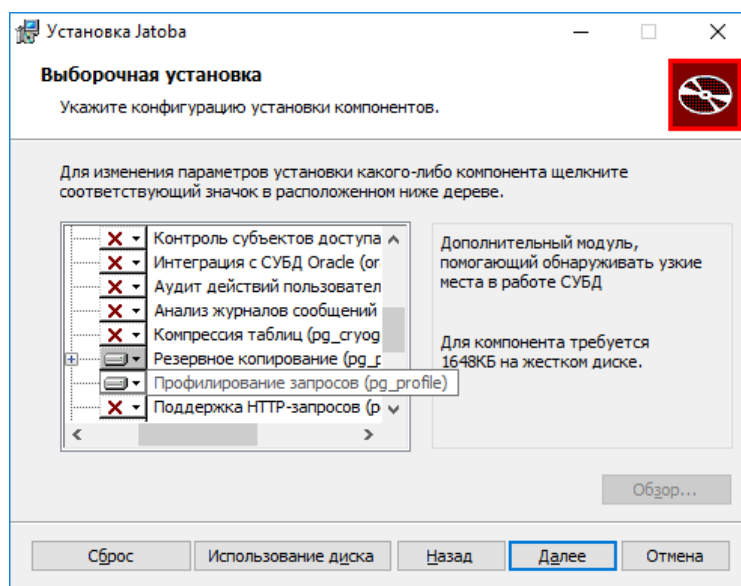


Рисунок 2.2 – Выбор устанавливаемых компонент

в) в открывшемся окне «Все готово к установке Jatoba» запустить процесс установки, нажав кнопку «Установить» (см. рис. 2.3);

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

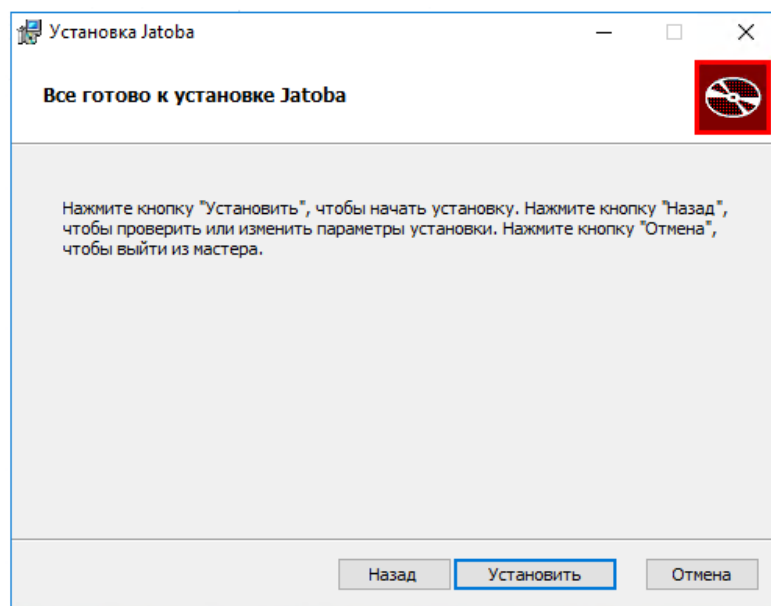


Рисунок 2.3 – Окно «Все готово к установке Jatoba»

2.2. Установка компонента ОС GNU/Linux

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

Установку компонента возможно провести двумя способами:

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

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

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

```
apt-get install jatoba<ver>-pg-profile
```

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

```
yum install jatoba<ver>-pg-profile
```

Отдельного уточнения требуют операционные системы ALT Linux и openSUSE.

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

```
apt-get install jatoba<ver>-pg-profile
```

– openSUSE также распространяется в виде rpm-пакетов, но использует собственный пакетный менеджер zypper, для нее команда установки выглядит следующим образом:

```
zypper install jatoba<ver>-pg-profile
```

Установка компонента в составе других версий СУБД «Jatoba» осуществляется аналогично. Отличие будет только в номере версии СУБД, в составе которой он распространяется. Например, jatoba3-pg-profile и т.п.

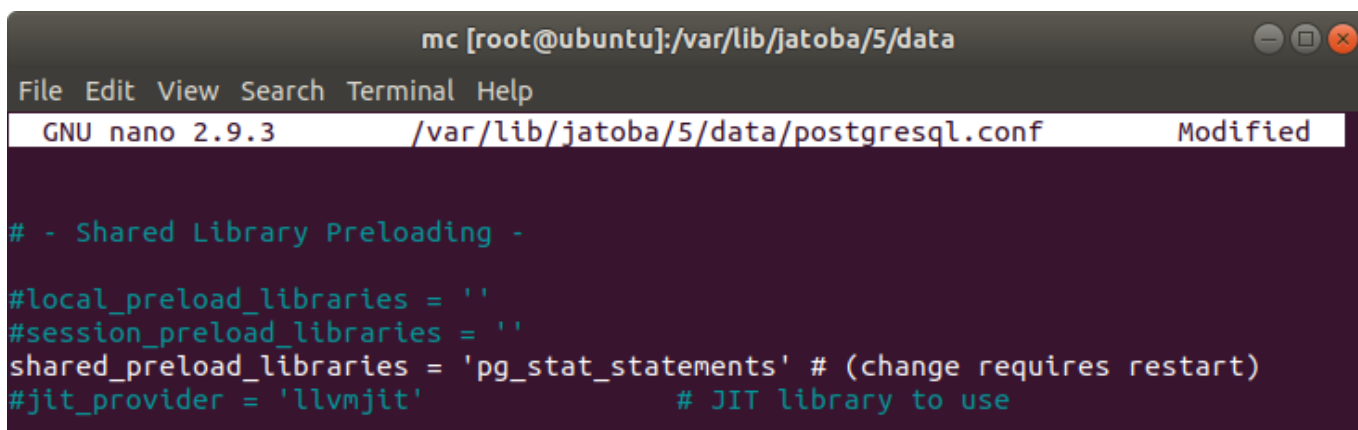
Удаление модуля также осуществляется средствами пакетного менеджера ОС. Вместо команды install нужно использовать соответствующую данному пакетному менеджеру команду удаления (remove, purge, erase и т.п.).

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

2.3. Рекомендованные настройки postgresql.conf

Для корректного функционирования расширения в разделе «Shared Library Preloading» в конфигурационном файле postgresql.conf необходимо добавить строку (рисунок 2.4):

```
shared_preload_libraries = 'pg_stat_statements'
```



```
mc [root@ubuntu]:/var/lib/jatoba/5/data
File Edit View Search Terminal Help
GNU nano 2.9.3 /var/lib/jatoba/5/data/postgresql.conf Modified

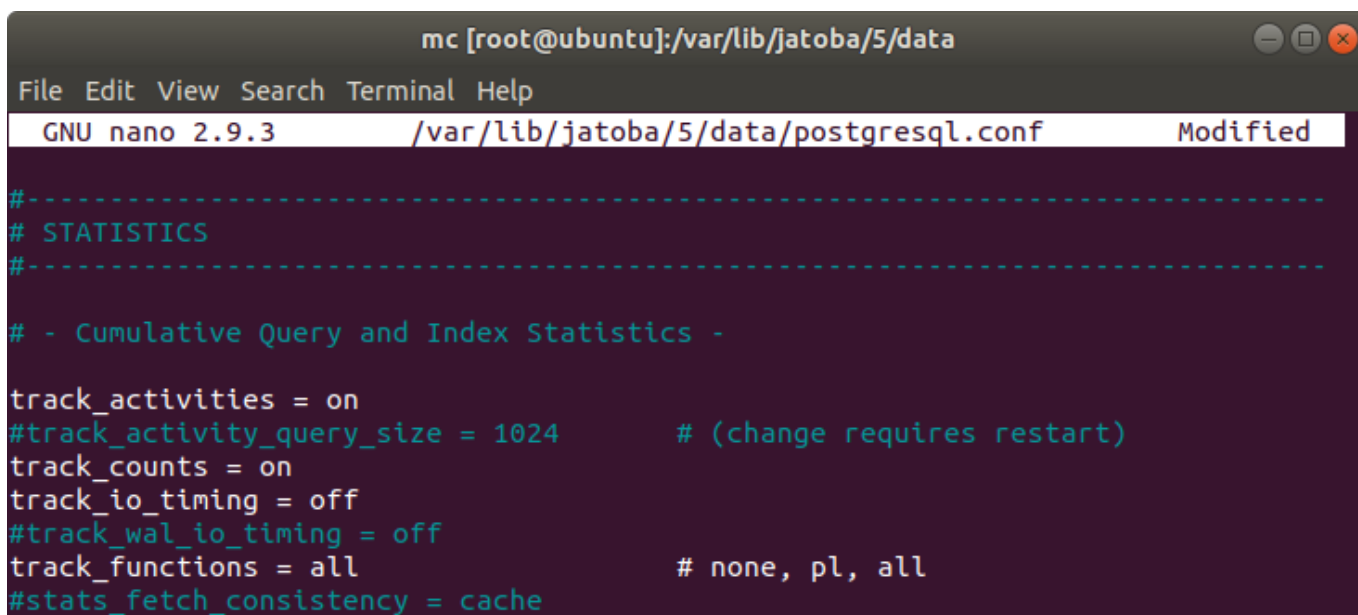
# - Shared Library Preloading -

#local_preload_libraries = ''
#session_preload_libraries = ''
shared_preload_libraries = 'pg_stat_statements' # (change requires restart)
#jit_provider = 'llvmjit' # JIT library to use
```

Рисунок 2.4 – Настройки postgresql.conf

В разделе «Statistics» – «Query and Index Statistics Collector» рекомендовано проверить и привести к следующему виду настройки для Statistic Collector (встроенный в СУБД механизм, позволяющий собирать метрики активности сервера БД) (рисунок 2.5):

```
track_activities = on
track_counts = on
track_io_timing = on
track_functions = all
```



```
mc [root@ubuntu]:/var/lib/jatoba/5/data
File Edit View Search Terminal Help
GNU nano 2.9.3 /var/lib/jatoba/5/data/postgresql.conf Modified

#-----
# STATISTICS
#-----

# - Cumulative Query and Index Statistics -

track_activities = on
#track_activity_query_size = 1024 # (change requires restart)
track_counts = on
track_io_timing = off
#track_wal_io_timing = off
track_functions = all # none, pl, all
#stats_fetch_consistency = cache
```

Рисунок 2.5 – Настройки postgresql.conf



В настройке `track_function` допустимым значением является «pl», которое отслеживает время выполнения функций, написанных на процедурных языках.

Значение «all» отслеживает все функции, к процедурным языкам добавляются SQL и C-функции.

После внесения изменений в конфигурационный файл, для применения настроек необходимо перезагрузить СУБД.

2.4. Установка расширения pg_Profile

Установка основного расширения «pg_profile» и дополнительных расширений допустима в:

- БД по умолчанию «postgres» (п. 2.4.1);
- служебную БД «pg_profile» (п. 2.4.2).

Для дальнейшего использования с компонентом пользовательского веб-интерфейса для администраторов «Jatoba data safe» целесообразнее устанавливать расширения в служебную БД, с последующей установкой внешнего сервера для служебной БД (п. 2.4.3).

2.4.1. Установка расширений в БД «postgres»

Установка расширения может быть проведена следующим способом:

```
postgres=# CREATE EXTENSION dblink;
```

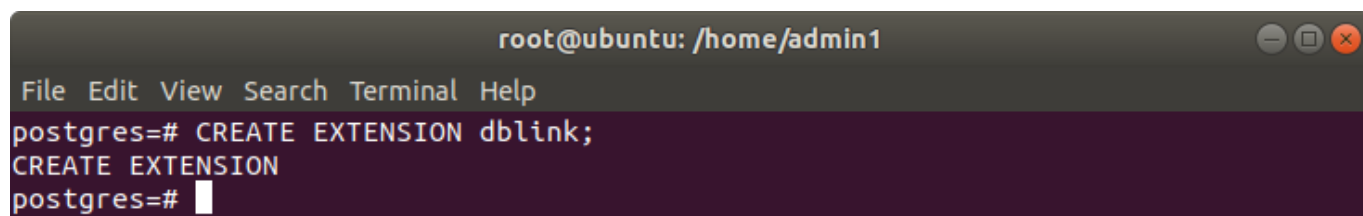


Рисунок 2.6 – Выполнение SQL-команды CREATE EXTENSION dblink

```
postgres=# CREATE EXTENSION pg_stat_statements;
postgres=# CREATE EXTENSION pg_profile;
```

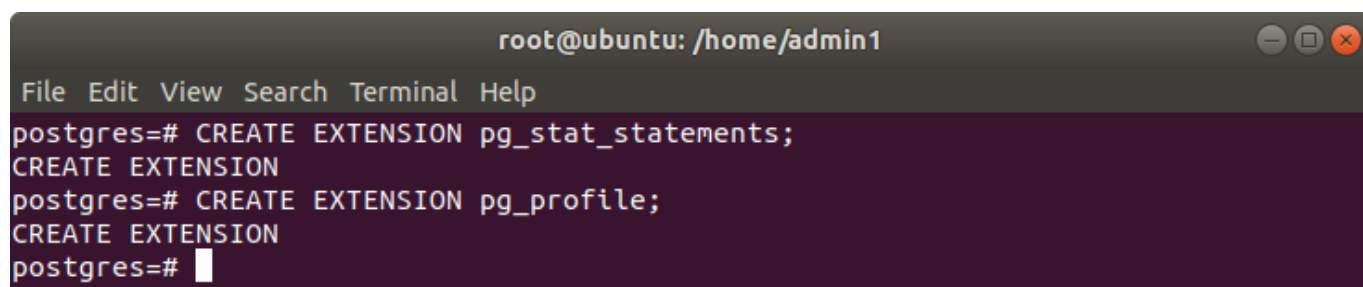
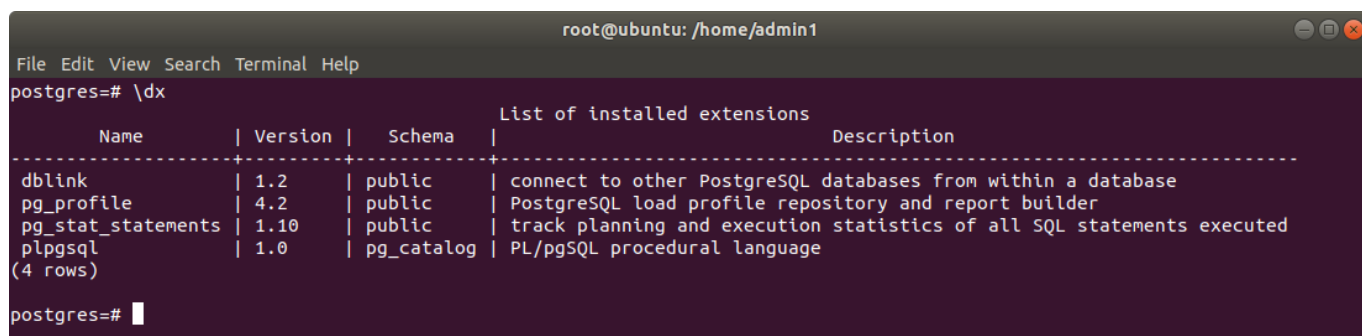


Рисунок 2.7 – Выполнение SQL-команд создания расширений

После выполнения данных команд будут созданы расширения СУБД (рисунок 2.8).



```

root@ubuntu: /home/admin1
File Edit View Search Terminal Help
postgres=# \dx

```

Name	Version	Schema	Description
dblink	1.2	public	connect to other PostgreSQL databases from within a database
pg_profile	4.2	public	PostgreSQL load profile repository and report builder
pg_stat_statements	1.10	public	track planning and execution statistics of all SQL statements executed
plpgsql	1.0	pg_catalog	PL/pgSQL procedural language

(4 rows)

```

postgres=#

```

Рисунок 2.8 – Созданные расширения СУБД

Также есть возможность создать расширение в отдельной схеме:

```

postgres=# CREATE EXTENSION dblink;
postgres=# CREATE EXTENSION pg_stat_statements;
postgres=# CREATE SCHEMA profile;
postgres=# CREATE EXTENSION pg_profile SCHEMA profile;

```

2.4.2. Установка расширений в служебную БД

В случае, когда архитектура безопасности информационной системы следует принципу назначения минимально необходимых прав и привилегий пользователям и администраторам, использование компонента от имени и с правами суперпользователя становится невозможным. Кроме того, БД «postgres» является БД по умолчанию и использование ее для установки расширений нежелательно.

Существует наиболее безопасный способ использования компонента «pg_Profile» с установкой в служебную БД, в которой устанавливаются требуемые расширения, правами использования которых обладает пользователь СУБД с минимально достаточными правами.

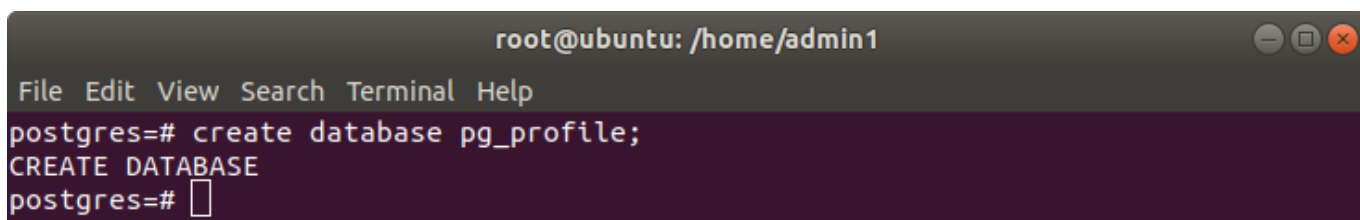
Для этого необходимо проделать следующие шаги:

- создать БД «pg_profile»:

```

create database pg_profile;

```

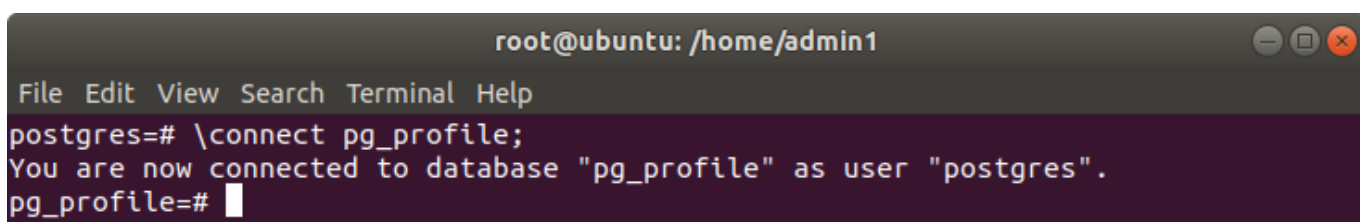


```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
postgres=# create database pg_profile;
CREATE DATABASE
postgres=#
```

Рисунок 2.9 – Создание БД «pg_profile»

- подключиться к БД:

```
\connect pg_profile;
```

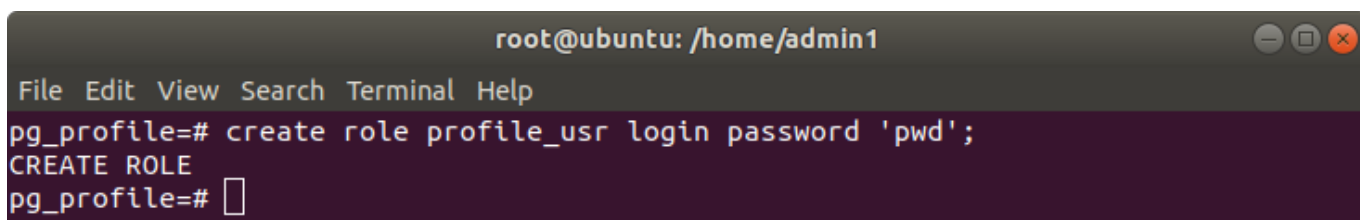


```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
postgres=# \connect pg_profile;
You are now connected to database "pg_profile" as user "postgres".
pg_profile=#
```

Рисунок 2.10 – Подключение к БД «pg_profile»

- создать пользователя «profile_usr»:

```
create role profile_usr login password 'pwd';
```

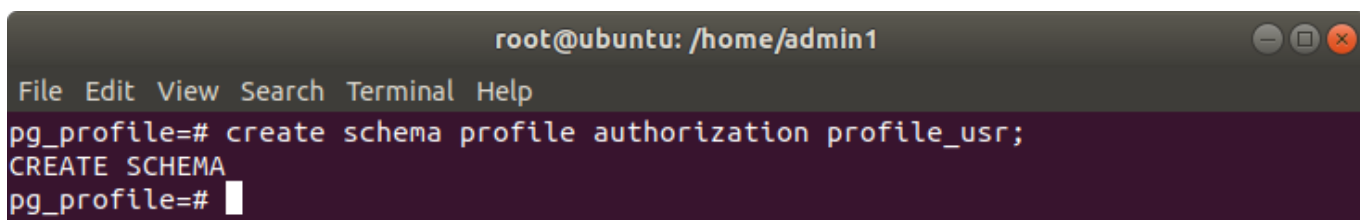


```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# create role profile_usr login password 'pwd';
CREATE ROLE
pg_profile=#
```

Рисунок 2.11 – Создание пользователя «profile_usr»

- создать схему для установки pg_profile:

```
create schema profile authorization profile_usr;
```

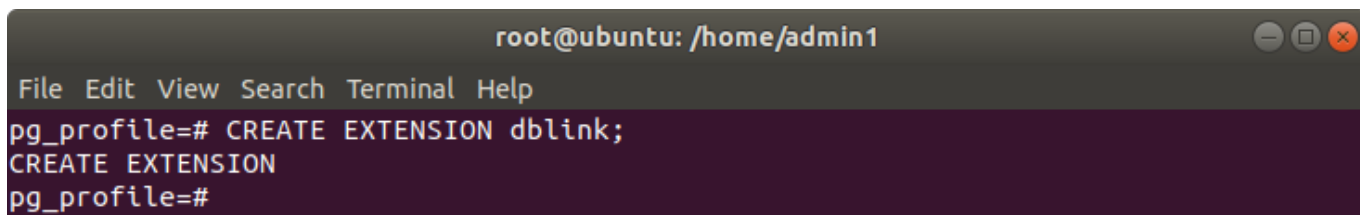


```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# create schema profile authorization profile_usr;
CREATE SCHEMA
pg_profile=#
```

Рисунок 2.12 – Создание схемы «profile»

- установить расширение «dblink»:

```
CREATE EXTENSION dblink;
```

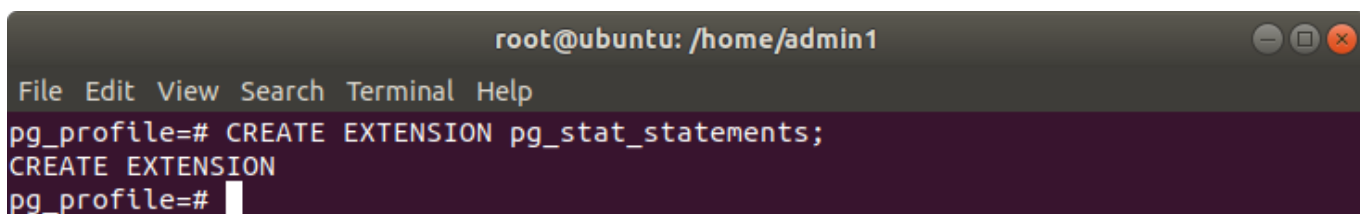


```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# CREATE EXTENSION dblink;
CREATE EXTENSION
pg_profile=#
```

Рисунок 2.13 – Установка расширения «dblink»

- установить расширение «pg_stat_statements»:

```
CREATE EXTENSION pg_stat_statements;
```

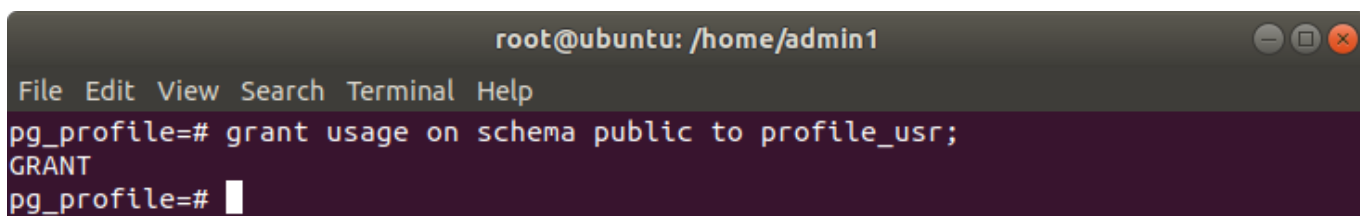


```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# CREATE EXTENSION pg_stat_statements;
CREATE EXTENSION
pg_profile=#
```

Рисунок 2.14 – Установка расширения «pg_stat_statements»

- предоставить разрешение на использование схемы public, в которой находится расширение dblink:

```
grant usage on schema public to profile_usr;
```



```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# grant usage on schema public to profile_usr;
GRANT
pg_profile=#
```

Рисунок 2.15 – Разрешение на использование схемы public



Предполагается, что расширение dblink установлено в схему public


```

root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# \dx

```

Name	Version	Schema	Description
dblink	1.2	public	connect to other PostgreSQL databases from within a database
pg_stat_statements	1.10	public	track planning and execution statistics of all SQL statements executed
plpgsql	1.0	pg_catalog	PL/pgSQL procedural language

```

(3 rows)
pg_profile=#

```

Рисунок 2.16 – Список установленных расширений в схемах данных

- создать расширение, «pg_profile» в схеме данных «profile»:

```
create extension pg_profile schema profile;
```

```

root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# create extension pg_profile schema profile;
CREATE EXTENSION
pg_profile=#

```

Рисунок 2.17 – Создание расширения, используя учетную запись profile_usr

На этом шаге установка компонента в отдельной БД закончена.

2.4.3. Установка внешнего сервера для служебной БД

Компонент обладает функциональной возможностью предоставления внешнего подключения пользователю или приложению. Для этого используется виртуальный сервер.

В разбираемом примере:

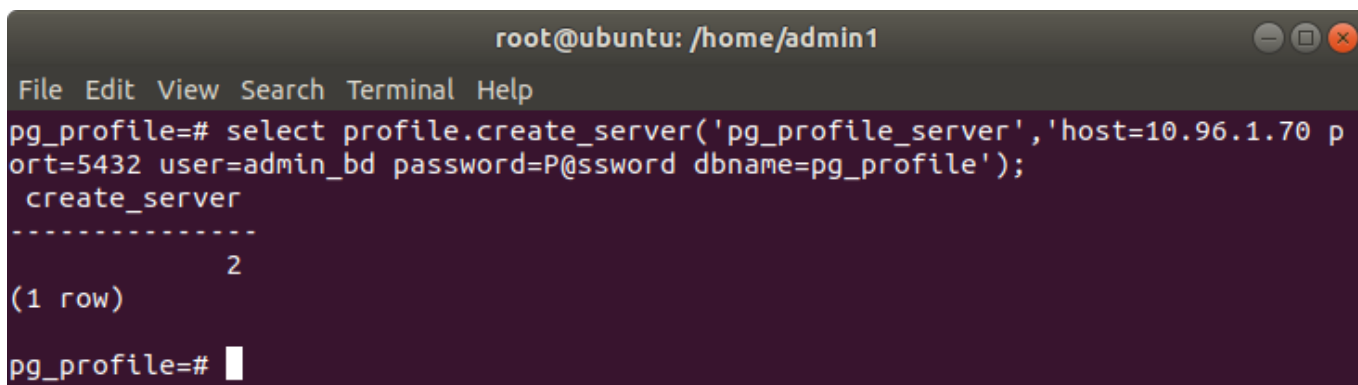
- создана служебная БД «pg_profile»;
- в служебной БД «pg_profile» установлены требуемые расширения;
- сервер СУБД имеет IP-адрес 10.96.1.70;
- создан пользователь «admin_bd» с атрибутом «Login».



В синтаксисе SQL-команды при вызове функций следует учитывать, что они установлены в отдельной схеме данных и для их вызова потребуется указывать схему данных, в которых они установлены.

Имея исходные данные, создать внешний сервер при помощи SQL-команды:

```
select  
profile.create_server('pg_profile_server','host=10.96.1.70  
port=5432 user=admin_bd password=P@ssword dbname=pg_profile');
```



```
root@ubuntu: /home/admin1  
File Edit View Search Terminal Help  
pg_profile=# select profile.create_server('pg_profile_server','host=10.96.1.70 p  
ort=5432 user=admin_bd password=P@ssword dbname=pg_profile');  
create_server  
-----  
2  
(1 row)  
pg_profile=#
```

Рисунок 2.18 – Создание внешнего сервера

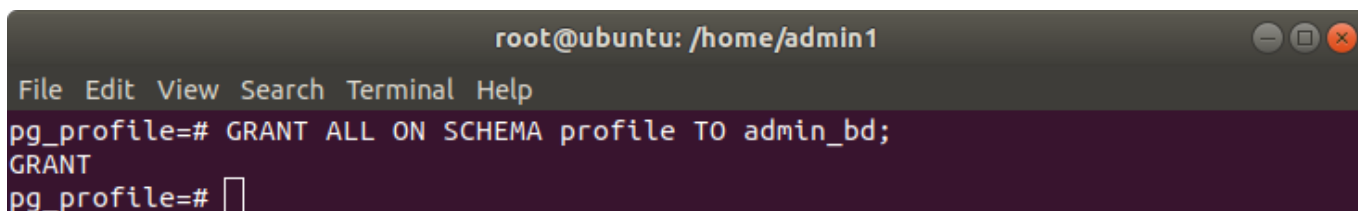
После выполнения SQL-команды в таблице «server» будет создана запись с строкой подключения внешнего пользователя СУБД.

Пользователю СУБД от имени и с правами которого будет производится подключение, достаточно иметь атрибут «Login» и дополнительные привилегии.

Для этого, необходимо назначить права на:

- схему данных «profile» SQL-командой:

```
GRANT ALL ON SCHEMA profile TO [username];
```

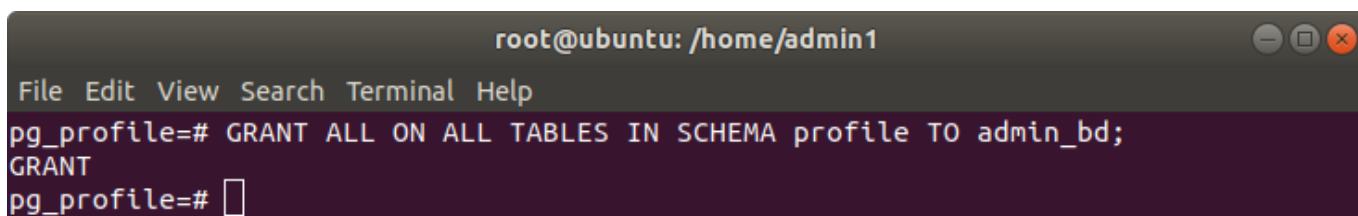


```
root@ubuntu: /home/admin1  
File Edit View Search Terminal Help  
pg_profile=# GRANT ALL ON SCHEMA profile TO admin_bd;  
GRANT  
pg_profile=#
```

Рисунок 2.19 – Предоставление прав на схему данных

- таблицы схемы данных «profile» SQL-командой:

```
GRANT ALL ON ALL TABLES IN SCHEMA profile TO [username];
```

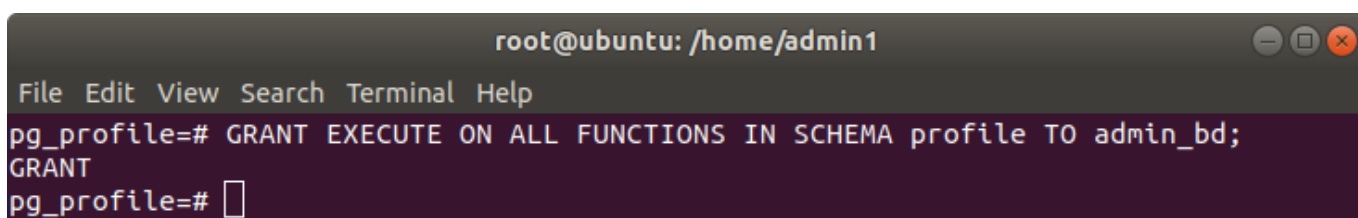


```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# GRANT ALL ON ALL TABLES IN SCHEMA profile TO admin_bd;
GRANT
pg_profile=#
```

Рисунок 2.20 – Предоставление прав на таблицы схемы данных «profile»

- функции схемы данных «profile» SQL-командой:

```
GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA profile TO [username];
```



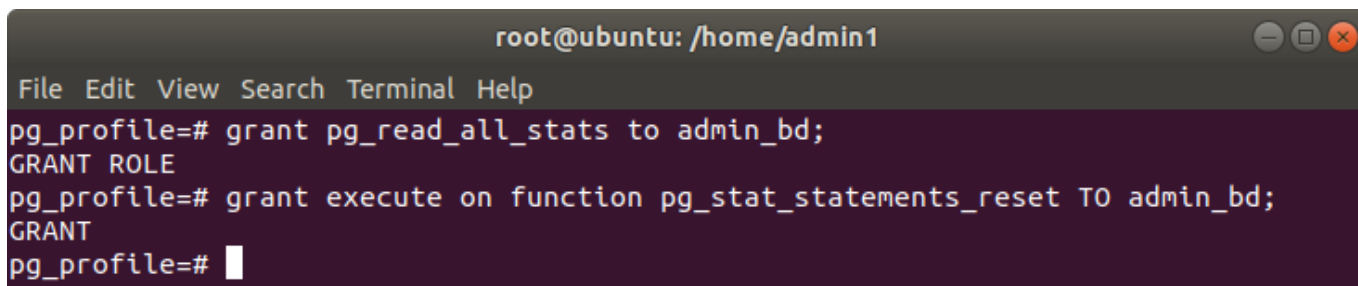
```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA profile TO admin_bd;
GRANT
pg_profile=#
```

Рисунок 2.21 – Предоставление прав на функции схемы «profile»

Необходимо убедиться, что пользователь имеет права на подключение к любой БД в СУБД (по умолчанию это так) и в конфигурационном файле «pg_hba.conf» прописаны разрешения для подключения с узла БД «pg_profile».

Кроме того, необходимо, чтобы пользователь СУБД от имени и с правами которого производится подключение, был включен в групповую роль «pg_read_all_stats» и имел привилегию «execute» на функцию «pg_stat_statements_reset».

```
# GRANT pg_read_all_stats to [username];
# GRANT execute on function pg_stat_statements_reset TO
[username];
```



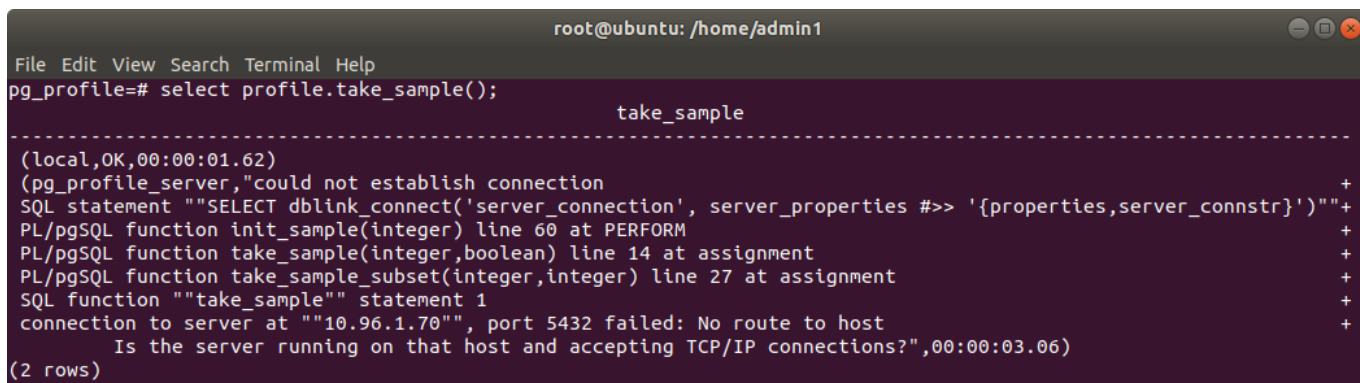
```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# grant pg_read_all_stats to admin_bd;
GRANT ROLE
pg_profile=# grant execute on function pg_stat_statements_reset TO admin_bd;
GRANT
pg_profile=#
```

Рисунок 2.22 – Включение в групповую роль «pg_stat_statements_reset» и назначение привилегии «execute» на функцию «pg_stat_statements_reset»

По умолчанию, вновь созданный сервер будет неактивен. Это отразится при просмотре состояния снапшотов:

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

```
SELECT profile.take_sample();
```

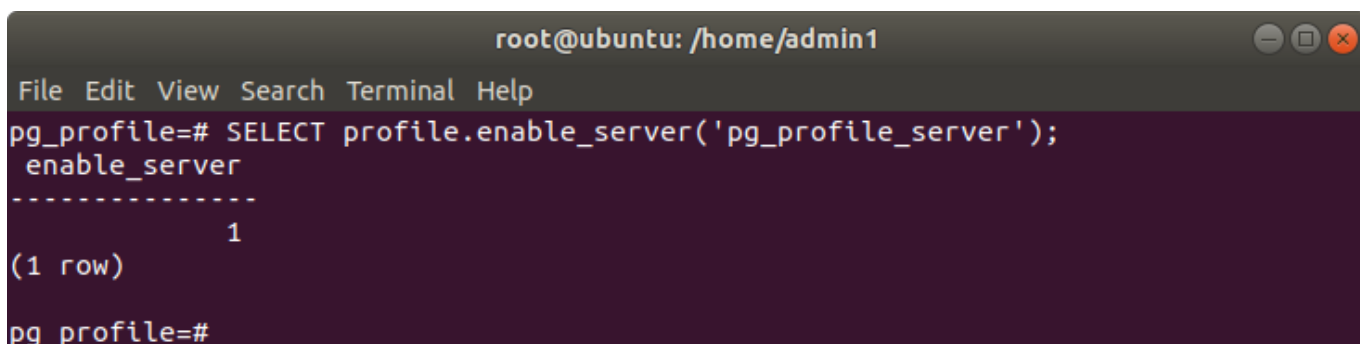


```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# select profile.take_sample();
               take_sample
-----
(local,OK,00:00:01.62)
(pg_profile_server,"could not establish connection
SQL statement ""SELECT dblink_connect('server_connection', server_properties #>> '{properties,server_connstr}')"
PL/pgSQL function init_sample(integer) line 60 at PERFORM
PL/pgSQL function take_sample(integer,boolean) line 14 at assignment
PL/pgSQL function take_sample_subset(integer,integer) line 27 at assignment
SQL function ""take_sample"" statement 1
connection to server at ""10.96.1.70"", port 5432 failed: No route to host
Is the server running on that host and accepting TCP/IP connections?",00:00:03.06)
(2 rows)
```

Рисунок 2.23 – Состояние снятия снапшотов

Включение сервера выполняется командой:

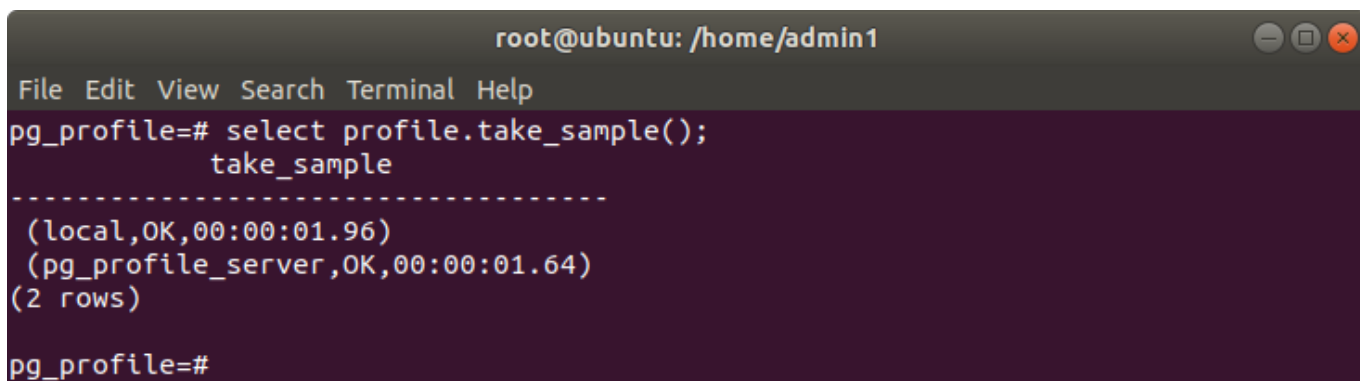
```
SELECT profile.enable_server('pg_profile_server');
```



```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# SELECT profile.enable_server('pg_profile_server');
           enable_server
-----
1
(1 row)
pg_profile=#
```

Рисунок 2.24 – Включение сервера с именем 'pg_profile_server'

После чего оба сервера будут активны, но для снижения нагрузки на СУБД, рекомендуется оставлять работающим один сервер.



```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# select profile.take_sample();
               take_sample
-----
(local,OK,00:00:01.96)
(pg_profile_server,OK,00:00:01.64)
(2 rows)
pg_profile=#
```

Рисунок 2.25 – Активное состояние серверов

На этом формирование подключения к внешнему серверу закончено.

2.5. Дополнительные настройки «postgresql.conf»

По умолчанию в postgresql.conf устанавливаются следующие параметры, которые можно поменять вручную:

- pg_profile.topn = 20 – количество основных объектов для каждого типа, которые будут выводиться в отчете;
- pg_profile.max_sample_age = 7 – время жизни снимка (снимки состояния, которые старше 7 дней будут удаляться при создании нового);
- pg_profile.track_sample_timings = off – настройка записи детальных данных о времени в снимок;
- pg_profile.max_query_length = 20000 – ограничение длины запроса для отчетов. Все запросы в отчете будут усечены до этой длины. Этот параметр не влияет на сбор текста запроса – во время выборки собираются полные тексты запросов, таким образом, они могут быть получены.

3. ИСПОЛЬЗОВАНИЕ

3.1. Управление соединениями с серверами

После установки расширение автоматически создаст соединение с локальным сервером, на котором оно установлено. Для управления другими соединениями имеются следующие функции:

```
create_server(server name, server_connstr text, server_enabled  
boolean = TRUE, max_sample_age integer = NULL, description text  
= NULL)
```

– создание нового соединения с сервером со следующими аргументами:

- `server` – уникальное имя сервера;
- `server_connstr` – строка соединения с сервером;
- `enabled` – включение сервера в набор серверов, на которых снимаются метрики при создании снапшота;
- `max_sample_age` – время жизни снапшота на сервере;
- `description` – описание сервера, которое будет указано в отчетах.

```
drop_server(server name)
```

– удаление сервера из настроек и удаление его снапшотов;

```
enable_server(server name)
```

– включает сервер в набор серверов, на которых снимаются метрики при создании снапшота;

```
disable_server(server name)
```

– исключает сервер из набора серверов, на которых снимаются метрики при создании снапшота;

```
rename_server(server name, new_name name)
```

– переименование сервера;

```
set_server_max_sample_age(server name, max_sample_age integer)
```

– определение времени жизни снапшота на сервере;



Функция `set_server_max_sample_age` перезаписывает значение `pg_profile.max_sample_age` в `postgresql.conf` для сервера.

Для отмены времени жизни снапшота, необходимо установить значение NULL.

```
set_server_db_exclude(server name, exclude_db name[])
```

– исключение из снапшота определенных БД на сервере (не будут указаны в отчетах);

```
set_server_connstr(server name, new_connstr text)
```

– установка правил соединения с сервером;

```
set_server_description(server name, description text)
```

– описание сервера (будет указано в отчетах);

```
show_servers()
```

– отображение созданных соединений с серверами.

Пример создания сервера:

```
SELECT profile.create_server('omega', 'host=name_or_ip  
dbname=postgres port=5432')
```

3.2. Сбор данных о размерах страниц

Для определения сбора данных о размерах таблиц реализована политика сбора данных.

Сбор данных о размерах таблиц в СУБД ресурсозатратный процесс и требует, чтобы таблица была эксклюзивно заблокирована на момент подсчета размера.



Данные о размерах таблиц можно собирать не постоянно, а, например, раз в сутки в выделенный промежуток времени.

Также можно установить временной интервал между двумя снимками состояния, где просчитываются данные о размерах таблиц.

– Функция `set_server_size_sampling()` определяет данные политики сбора размеров.

```
set_server_size_sampling(server name, window_start time with  
time zone = NULL, window_duration interval hour to second =  
NULL, sample_interval interval day to minute = NULL)
```

`server` – имя сервера;

`window_start` – начало временного промежутка, когда разрешено собирать данные о размерах таблиц;

`window_duration` – продолжительность временного промежутка;

`sample_interval` – минимальное время между двумя снимками с собранными размерами таблиц.



Все три параметра являются обязательными для установки политики сбора данных о размерах таблиц.

Пример:

```
SELECT set_server_size_sampling('local','23:00+03', interval '2  
hour',interval '8 hour')
```

– Функция `show_servers_size_sampling()` показывает все установленные политики.

3.3. Снимки состояния БД

Каждый снимок показывает статистическую информацию о метриках БД и нагрузке с момента снятия предыдущего состояния.

Для работы со снимками используются следующие команды:

```
take_sample(server name [, skip_sizes boolean])
```

– снятие снимков со всех серверов.

server_name – имя сервера для снятия снапшота;

skip_sizes – переопределение установленной политики сбора данных о размерах.



Снапшоты для каждого сервера будут сняты последовательно, один за другим.

Если не указывать параметры server_name, то будут сняты снапшоты со всех доступных серверов.

При запросе функции без указания параметров пользователю будет возвращена таблица вида (см. рис. 3.1):

```
root@ubuntu: /home/admin1
File Edit View Search Terminal Help
pg_profile=# select profile.take_sample();
           take_sample
-----
(local,OK,00:00:01.5)
(1 row)
pg_profile=#
```

Рисунок 3.1 – Возвращение таблицы

server name – имя сервера;

result text – результат (ОК – успешно, текст ошибки – неудача);

elapsed interval – время, затраченное на создание снапшота.



Работа функции take_sample() может занять значительное время из-за последовательного снятия снапшотов.

```
take_sample_subset([sets_cnt integer], [current_set integer])
```

– функция take_sample() обрабатывает данные и создает снапшоты последовательно. В случае, если пользователь регистрирует множество серверов через функцию create_server(), то для получения всех снапшотов может быть затрачено значительное время. Функция take_sample_subset() предоставляет возможность использовать параллельную обработку серверов, тем самым, сокращая время на получение всех снапшотов.

`sets_cnt` – число, на которое будет разделено все множество серверов. К примеру, если у пользователя есть 20 серверов, то указывая `sets_cnt = 5`, будет создано 5 подмножеств по 4 сервера.

`current_set` – номер подмножества, которое будет обрабатываться в данный момент. Счет идет от 0, максимальное значение `current_set = sets_cnt – 1`.

Можно запускать как один за другим, контролируя таким образом время выполнения, так и запустить параллельно в нескольких сеансах, указывая разный `current_set`.

```
show_samples([server name,] [days integer])
```

– возвращение таблицы с данными об имеющихся снапшотах. Если не вводить имя сервера, то автоматически подставится локальный сервер.

- `sample integer` – ID снапшота;
- `sample_time timestamp (0) with time zone` – время, когда был снят снапшот;
- `dbstats_reset / clustats_reset / archstats_reset timestamp (0) with time zone` – время сброса статистики в представлениях `pg_stat_database`, `pg_stat_bgwriter` с `pg_stat_archiver` с момента последнего снятия снапшота.

3.3.1. Снятие снапшотов

Для построения отчета необходимо, как минимум, 2 снапшота.

Для автоматического получения снимков по расписанию, к примеру, раз в 30 минут, нужно поместить вызов функции в планировщик задач системы.

В стандартной утилите Linux – `crontab`:

```
*/30 * * * * psql -c 'SELECT [profile].take_sample();' >  
/dev/null 2>&1
```

В MS Windows (начиная с Windows Server 2003) – стандартной утилитой `schtasks`:

```
schtasks /create /sc minute /mo 30 /tn "30_min_sample" /tr psql  
-c "SELECT [profile].take_sample();"
```

Для получения списка существующих снапшотов в хранилище используется функция `show_samples()`. Эта функция также покажет обнаруженные сбросы статистики БД.

3.3.2. Время жизни снимков

Для определения срока хранения снимков реализована политика хранения. Определить политику хранения можно на трех уровнях:

- 1) установка параметра `pg_profile.max_sample_age` в файле `postgresql.conf` – общая настройка, которая действует, если не определено ни одно из других;
- 2) определение параметра `max_sample_age` сервера при создании сервера или использование функции `set_server_max_sample_age()` для существующего сервера – настройка переопределяет глобальную настройку `pg_profile.max_sample_age` для конкретного сервера;
- 3) создание `baseline` (см. пункт 3.4) – `baseline` будет переопределять период хранения для включенных снимков с наивысшим приоритетом.

3.3.3. Описание событий

`pg_profile` собирает подробную статистику времени снятия снимков, если включен параметр `pg_profile.track_sample_timings`. Результаты можно получить из представления `v_sample_timings`.

Описание полей `v_sample_timings`:

- `server_name` – имя сервера с которого создавался снимок;
- `sample_id` – идентификатор снимка;
- `sample_time` – время взятия снимка;
- `event` – событие на котором измерялось время;
- `time_spent` – количество времени, проведенного в событии.

В таблице 3.1 описаны все события и метрики, которые доступны пользователю в созданном отчете.

Таблица 3.1 – Описание событий и метрик

Описание	Значение
total	снятие снимка (все этапы)
connect	установка соединения <code>dblink</code> с сервером
get server environment	получение конфигурационных параметров сервера, доступных расширений и т.д.

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

Описание		Значение
collect database stats		запрос представления pg_stat_database для получения статистики по БД
calculate database stats		вычисление дифференциальной статистики по БД с момента предыдущего снимка
collect tablespace stats		запрос представления pg_tablespace для получения статистики по табличным пространствам
collect statement stats		сбор статистики по операторам с помощью расширений pg_stat_statements и pg_stat_kcache
query pg_stat_bgwriter		сбор статистики кластера с помощью представления pg_stat_bgwriter
query pg_stat_archiver		сбор статистики кластера с помощью представления pg_stat_archiver
collect object stats		сбор статистики по объектам БД. Включает события
	db:dbname collect tables stats	сбор статистики по таблицам для БД dbname
	db:dbname collect indexes stats	сбор статистики по индексам для БД dbname
	db:dbname collect functions stats	сбор статистики по функциям для БД dbname
maintain repository		выполнение процедур поддержки
calculate tablespace stats		вычисление дифференциальной статистики по табличным пространствам. Включает события
	calculate object stats	вычисление дифференциальной статистики по объектам БД
	calculate tables stats	вычисление дифференциальной статистики по таблицам БД
	calculate indexes stats	вычисление дифференциальной статистики по индексам БД
calculate functions stats		вычисление дифференциальной статистики по функциям БД
calculate cluster stats		вычисление дифференциальной статистики кластера
calculate archiver stats		вычисление дифференциальной статистики архиватора
delete obsolete samples		удаление устаревших baseline и снимков

3.4. Baseline

Baseline – именованная последовательность снимков, которая имеет отдельную от настроенной политику хранения. Можно задать как определенное время хранения в днях, так и бесконечное время хранения, оставив соответственный параметр пустым. Также можно создать последовательность снимков только для определенного периода времени.

Можно сохранить снимки, собранные во время нагрузочного тестирования или во время обычной нагрузки на систему для дальнейшего использования.

3.4.1. Функции управления baseline

Для управления baseline предназначены следующие функции:

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

```
CREATE_baseline([server name,] baseline_name varchar(25),  
start_id integer, end_id integer [, days integer])
```

– создание baseline,

server name – имя сервера (если параметр опущен, предполагается локальный сервер);

baseline_name – имя baseline. Каждый baseline должен иметь уникальное имя в пределах сервера;

start_id, end_id – первый и последний образцы, включенные в baseline;

days – время хранения baseline. Определяется в целых днях с момента времени now(). Этот параметр может быть опущен (или установлен в null), что означает неограниченное хранение.

```
CREATE_baseline([server name,] baseline_name varchar(25),  
time_range tstzrange [, days integer])
```

– создание baseline,

server name – имя сервера (если параметр опущен, предполагается локальный сервер);

baseline_name – имя baseline. Каждый baseline должен иметь уникальное имя в пределах сервера;

time_range – временной интервал для baseline. Baseline будет включать все доступные снимки, перекрывающие этот интервал;

days – время хранения baseline. Определяется в целых днях с момента времени now(). Этот параметр может быть опущен (или иметь значение null), что означает неограниченное хранение.

```
drop_baseline([server name,] name varchar(25))
```

– удаление baseline,

server name – имя сервера (если параметр опущен, предполагается локальный сервер);

name – имя baseline для удаления. Удаление baseline не означает удаление его снимков, они исключаются из baseline.

```
keep_baseline([server name,] name varchar(25) [, days integer])
```

– изменение времени хранения baseline,

server name – имя сервера (если параметр опущен, предполагается локальный сервер);

name – имя baseline. Каждый baseline должен иметь уникальное имя в пределах сервера;

days – время хранения baseline. Определяется в целых днях с момента времени now(). Этот параметр может быть опущен (или иметь значение null), что означает неограниченное сохранение.

```
show_baselines([server name])
```

– просмотр существующих baseline и их информация,

server name – имя сервера (если параметр опущен, предполагается локальный сервер).

Пример просмотра существующих baseline:

```
postgres=# SELECT * FROM show_baselines('local');
```

3.5. Экспорт / импорт данных

Собранные снапшоты можно экспортировать из экземпляра расширения pg_Profile и импортировать на другой сервер.

3.5.1. Экспорт данных

Данные экспортируются в виде обычной таблицы с помощью функции export_data(). Можно использовать любой метод для экспорта этой таблицы из БД. Например, можно использовать команду copy в psql для получения одного файла .csv:

```
postgres=# \copy (select * from export_data()) to 'export.csv'
```

По умолчанию функция export_data() экспортирует все снапшоты всех настроенных серверов. Можно ограничить экспорт только одним сервером, а также ограничить диапазон снапшотов:

```
export_data([server name, [min_sample_id integer,]  
[max_sample_id integer]] [, obfuscate_queries boolean])
```

– экспорт собранных данных,

`server` – имя сервера (если параметр опущен, предполагаются все серверы);

`min_sample_id` и `max_sample_id` – экспорт идентификаторов граничных снапшотов (включительно). Нулевое значение `min_sample_id` приводит к экспорту всех снапшотов до `max_sample_id`, а нулевое значение `max_sample_id` приводит к экспорту всех снапшотов, начиная с `min_sample_id`;

`obfuscate_queries` – скрытие текста запросов, они будут экспортированы в виде MD5-хэша.

3.5.2. Импорт данных

Данные могут быть импортированы только из локальной таблицы, поэтому ранее экспортированные данные необходимо сначала загрузить с помощью команды `copy`:

```
# CREATE TABLE import (section_id bigint, row_data json);  
# \copy import from 'export.csv'
```

После загрузки можно выполнить импорт данных, предоставив эту таблицу функции `import_data()`:

```
# SELECT * FROM import_data('import');
```



Одинаково названные `pg_profile` сервера на разных физических серверах при импорте данных не будут обработаны из-за внутренних ограничений функции. Во избежание данной ситуации нужно временно переименовать их через встроенную функцию `rename_server(server name, new_name name)`

Если понадобится импортировать новые данные для ранее импортированных серверов, они будут сопоставлены по системному идентификатору. Все серверы импортируются с опцией `server_enabled = FALSE` (см. пункт 3.1).

Функция `import_data()` принимает только импортируемую таблицу:

```
import_data(data regclass)
```

data – таблица, содержащая экспортируемые данные. Эта функция возвращает количество строк, фактически загруженных в таблицы расширения. После успешного завершения операции можно удалить таблицу импорта.

3.6. Создание отчетов

Отчеты создаются в формате HTML. Существует два вида отчетов: обычные отчеты и дифференциальные отчеты.

3.6.1. Обычные отчеты

Обычные отчеты содержат статистическую информацию для определенного периода времени.

Функции обычного отчета:

```
get_report([server name,] start_id integer, end_id integer [,  
description text [, with_growth boolean])
```

– создание отчета, исходя из идентификаторов снапшотов;

```
get_report([server name,] time_range tstzrange [, description  
text [, with_growth boolean])
```

– генерирование отчета для самого короткого интервала снапшотов, охватывающего указанный time_range;

```
get_report([server name], baseline varchar(25) [, description  
text [, with_growth boolean])
```

– генерирование отчета, используя baseline в качестве интервала снапшотов;

```
get_report_latest([server name])
```

– создание отчета, исходя из двух последних снапшотов,

server – имя сервера (если параметр опущен, предполагается локальный сервер);

start1_id, end1_id – идентификаторы снапшота первого интервала;

start2_id, end2_id – идентификаторы снапшота второго интервала;

baseline1 – имя baseline первого интервала;

baseline2 – имя baseline второго интервала;

time_range1 – временной диапазон первого интервала;

time_range2 – временной диапазон второго интервала;

description – текстовая заметка, которая будет включена в отчет, как описание отчета;

with_growth – параметр, который означает, что будет браться ближайший снимок с данными о размерах.

3.6.2. Дифференциальные отчеты

Дифференциальные отчеты содержат статистическую информацию за два выбранных периода, позволяя легко сравнить показатели.

Функции дифференциального отчета:

```
get_diffreport([server name,] start1_id integer, end1_id  
integer, start2_id integer, end2_id integer [, description text  
[, with_growth boolean]])
```

– создание отчета на основе двух периодов времени;

```
get_diffreport([server name,] baseline1 varchar(25), baseline2  
varchar(25) [, description text [, with_growth boolean]])
```

– создание дифференциального отчета по двум интервалам, заданными именами baseline;

```
get_diffreport([server name,] time_range1 tstzrange,  
time_range2 tstzrange [, description text [, with_growth  
boolean]])
```

– создание дифференциального отчета по двум интервалам, заданным временными диапазонами,

server – имя сервера (если параметр опущен, предполагается локальный сервер);

start1_id, end1_id – идентификаторы снимков первого интервала;

start2_id, end2_id – идентификаторы снимков второго интервала;

baseline1 – название базовой линии первого интервала;

baseline2 – название базовой линии второго интервала;

time_range1 – временной диапазон первого интервала;

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

time_range2 – временной диапазон второго интервала;

with_growth – параметр, который означает, что будет браться ближайший снимок с данными о размерах;

description – текстовая информация, которая будет выведена в отчете.

Пример формирования отчета:

```
$ psql -AqtC "SELECT profile.get_report('omega',12,14)" -o  
report_omega_12_14.html
```

4. ДАННЫЕ В ОТЧЕТАХ

В данном разделе описаны таблицы отчетов:

- 1) статистики сервера (Server statistics п. 4.1);
- 2) статистики SQL-запросов (SQL query statistics п. 4.2);
- 3) статистики использования ресурсов (Rusage statistics п. 4.3);
- 4) статистики объекта схемы (Schema object statistics п. 4.4);
- 5) статистики функций пользователя (User function statistics п. 4.5);
- 6) статистики, связанной с вакуумом (Vacuum-related statistics п. 4.6);
- 7) настройки кластера во время отчетного интервала (Cluster settings during the report interval п. 4.7);
- 8) Отчеты по компоненту «ja_Hipe_Cluster» (Citrus) (п. 4.8).

В каждом подразделе приведены подробные виды отчета.



Наименование таблиц в отчетах отображаются на английском языке.

4.1. Server statistics (Статистика сервера)

Данный раздел содержит сведения о метриках и статистических данных, собранных в генерируемом отчете.

4.1.1. Database statistics (Статистика базы данных)

Таблица 4.1 содержит статистику по БД в течение интервала отчета, основанную на представлении pg_stat_database.

Database	Transactions			Block statistics			Tuples					Temp files		Size	Growth
	Commits	Rollbacks	Deadlocks	Hit(%)	Read	Hit	Ret	Fet	Ins	Upd	Del	Size	Files		
jatoba	1101			99.40	182	30278	34803	6649	2039	3234	4			25 MB	1376 kB
jatoba	7			100.00		7663	9738	3021						10217 kB	
Total	1183			99.63	415	112798	128843	50850	4041	3838	16			50 MB	2264 kB

Рисунок 4.1 – Пример отчета Database statistics

Таблица 4.1 – Описание параметров отчета Database statistics

Параметр	Описание
Database	имя БД
Transaction	статистика транзакций БД

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

Параметр		Описание
	Commits	количество зафиксированных транзакций (xact_commit)
	Rollbacks	количество откатенных транзакций (xact_rollback)
	Deadlocks	количество обнаруженных взаимных блокировок (deadlocks)
Block statistics		статистика чтения и попаданий в блоки БД
	Hit(%)	коэффициент попадания в буферный кэш
	Read	количество прочитанных дисковых блоков в данной БД (blks_read)
	Hit	количество раз, когда дисковые блоки были найдены в буферном кэше (blks_hit)
Tuples		раздел статистики записей
	Ret	количество возвращенных записей (tup_returned)
	Fet	количество извлеченных записей (tup_fetched)
	Ins	количество вставленных записей (tup_inserted)
	Upd	количество обновленных записей (tup_updated)
	Del	количество удаленных записей (tup_deleted)
Temp files		статистика временных файлов
	Size	общий объем данных, записанных во временные файлы запросами в этой БД (temp_bytes)
	Files	количество временных файлов, созданных запросами в этой БД (temp_files)
Size		размер БД в конце интервала отчета (pg_database_size())
Growth		рост БД в течение интервала отчета (разница pg_database_size())

4.1.2. Statement statistics by database

Таблица 4.2 содержит агрегированные общие статистические данные по базе данных pg_stat_statements (если расширение pg_stat_statements было доступно в течение интервала отчета).

Database	Calls	Time (s)				Fetched (blk)		Dirtied (blk)		Temp (blk)		Local (blk)		Statements	WAL size
		Exec	Read	Write	Trg	Shared	Local	Shared	Local	Read	Write	Read	Write		
pg_profile	33	2.38				1664152		249						23	1629 kB
postgres	8	0.05				7415		1						8	
Total	41	2.44				1671567		250						31	1629 kB

Рисунок 4.2 – Пример отчета Statement statistics by database

Таблица 4.2 – Описание параметров отчета Statement statistics by database

Параметр		Описание
Database		имя БД
Calls		общее количество исполнений всех запросов (сумма вызовов).
Time (s)		затраченное время в секундах
	Plan	время, затраченное на планирование (сумма total_plan_time) – доступно с версии pg_stat_statements 1.8

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

Параметр		Описание
	Exec	время, затраченное на выполнение (сумма total_time или total_exec_time)
	Read	время, затраченное на чтение блоков (сумма blk_read_time)
	Write	время, затраченное на запись блоков (сумма blk_write_time)
	Trg	время, затраченное на выполнение функций триггера
Temp I/O Time		время, потраченное на временные блоки ввода-вывода файлов
	Read	время, затраченное на чтение временных блоков файлов, в секундах
	Write	время, затраченное на запись временных блоков файлов в секундах
Fetched (blk)		общее количество блоков, извлеченных с диска и кэша буфера
	Shared	общее количество найденных общих блоков (сумма shared_blks_read + shared_blks_hit)
	Local	общее количество локальных блоков (сумма local_blks_read + local_blks_hit)
Dirtied (blk)		общее количество блоков, заполненных в БД
	Shared	общее количество общих блоков, заполненных в БД (сумма shared_blks_dirtied)
	Local	общее количество локальных блоков, заполненных в БД (сумма local_blks_dirtied)
Temp (blk)		блоки, используемые для операций соединения и сортировки
	Read	прочитанные блоки (сумма temp_blks_read)
	Write	записанные блоки (сумма temp_blks_written)
Local (blk)		блоки, используемые для временных таблиц
	Read	прочитанные блоки (сумма local_blks_read)
	Write	записанные блоки (сумма local_blks_written)
Statements		общее количество перехваченных заявлений
WAL size		общий объем WAL, сгенерированный операторами (сумма wal_bytes)

4.1.3. Statement statistics by database (Статистика запросов по базе данных)

Таблица 4.3 содержит агрегированную общую статистику данных pg_stat_statements по каждой БД (если расширение pg_stat_statements было доступно в течение интервала отчета).

Database	Calls	Time (s)					Fetched (blk)		Dirtied (blk)		Temp (blk)		Local (blk)		Statements	WAL size
		Plan	Exec	Read	Write	Trg	Shared	Local	Shared	Local	Read	Write	Read	Write		
jatoba	8	0.47	0.16				28851								25	
jatoba	120	0.01	0.02				6872		5						4	17 kB
Total	128	0.49	0.91				100050		1038						51	8450 kB

Рисунок 4.3 – Пример отчета Statement statistics by database

Таблица 4.3 – Описание параметров отчета Statement statistics by database

Параметр		Описание
Database		имя БД
Calls		общее количество выполнения всех запросов (сумма вызовов)
Time(s)		затраченное время в секундах
	Plan	время, затраченное на планирование (сумма total_plan_time) – доступно с версии pg_stat_statements 1.8
	Exec	время, затраченное на выполнение (сумма total_time или total_exec_time)
	Read	время, затраченное на чтение блоков (сумма blk_read_time)
	Write	время, затраченное на запись блоков (сумма blk_write_time)
	Trg	время, затраченное на выполнение функций триггера
Fetched (blk)		общее количество блоков, извлеченных с диска и кэша буфера
	Shared	общее количество найденных общих блоков (сумма shared_blks_read + shared_blks_hit)
	Local	общее количество локальных блоков (сумма local_blks_read + local_blks_hit)
Dirtied (blk)		общее количество блоков, заполненных в БД
	Shared	общее количество общих блоков, заполненных в БД (сумма shared_blks_dirtied)
	Local	общее количество локальных блоков, заполненных в БД (сумма local_blks_dirtied)
Temp (blk)		блоки, используемые для операций соединения и сортировки
	Read	прочитанные блоки (сумма temp_blks_read)
	Write	записанные блоки (сумма temp_blks_written)
Local (blk)		блоки, используемые для временных таблиц
	Read	прочитанные блоки (сумма local_blks_read)
	Write	записанные блоки (сумма local_blks_written)
Statements		общее количество перехваченных заявлений
WAL size		общий объем WAL, сгенерированный операторами (сумма wal_bytes)

4.1.4. Cluster statistics (Статистика кластера)

Таблица 4.4 содержит данные из представления pg_stat_bgwriter.

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

Metric	Value
Scheduled checkpoints	
Requested checkpoints	
Checkpoint write time (s)	
Checkpoint sync time (s)	
Checkpoints buffers written	
Background buffers written	
Backend buffers written	
Backend fsync count	
Bgwriter interrupts (too many buffers)	
Number of buffers allocated	390
WAL generated	10145 kB
WAL segments archived	
WAL segments archive failed	

Рисунок 4.4 – Пример отчета Cluster statistics

Таблица 4.4 – Описание параметров отчета Cluster statistics

Параметр	Описание
Scheduled checkpoints	общее количество контрольных точек, завершенных по расписанию благодаря параметру checkpoint_timeout (поле checkpoints_timed)
Requested checkpoints	общее количество других контрольных точек из-за значений параметров max_wal_size, archive_timeout и команды CHECKPOINT (поле checkpoints_req)
Checkpoint write time(s)	общее время записи контрольных точек в секундах (поле checkpoint_write_time)
Checkpoint sync time(s)	общее время синхронизации контрольных точек в секундах (поле checkpoint_sync_time)
Checkpoints buffers written	общее количество буферов, записанных процессом Checkpointer (поле buffers_checkpoint)
Background buffers written	общее количество буферов, записанных фоновым процессом записи (поле buffers_clean)
Backend buffers written	общее количество буферов, записанных backend (поле buffers_backend)
Backend fsync count	общее количество вызовов fsync backend (поле buffers_backend_fsync)
Bgwriter interrupts (too many buffers)	общее количество прерываний фонового писателя из-за достижения значения параметра bgwriter_lru_maxpages
Number of buffers allocated	общее количество выделенных буферов (поле buffers_alloc)
WAL generated	общее количество сгенерированных WAL (на основе pg_current_wal_lsn())
WAL segments archived	количество архивированных сегментов WAL (на основе archived_count представления pg_stat_archiver)

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

Параметр	Описание
WAL segments archive failed	количество неудач архивирования сегментов WAL (на основе failed_count представления pg_stat_archiver)

4.1.5. WAL statistics (Статистика WAL)

Таблица 4.5 содержит данные из представления pg_stat_wal1.

WAL statistics

Metric	Value
WAL generated	1892 kB
WAL per second	21 kB
WAL records	7628
WAL FPI	176
WAL buffers full	
WAL writes	23
WAL writes per second	0.26
WAL sync	20
WAL syncs per second	0.22
WAL write time (s)	
WAL write duty	
WAL sync time (s)	
WAL sync duty	

Рисунок 4.5 – Пример отчета WAL statistics

Таблица 4.5 – Описание параметров отчета WAL statistics

Параметр	Описание
WAL generated	Общий объем сгенерированного WAL (wal_bytes)
WAL per second	Средний объем WAL, генерируемого в секунду
WAL records	Общее количество сгенерированных записей WAL (wal_records)
WAL FPI	Общее количество созданных полных изображений WAL-страниц (wal_fpi)
WAL buffers full	Количество раз, когда данные WAL записывались на диск из-за того, что буферы WAL заполнялись (wal_buffers_full)
WAL writes	Количество раз, когда буферы WAL записывались на диск с помощью запроса XLogWrite (wal_write)
WAL writes per second	Среднее количество раз, которое буферы WAL записываются на диск посредством запроса XLogWrite в секунду

¹ для версии компонента 4.2

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

Параметр	Описание
WAL sync	Количество раз, когда файлы WAL были синхронизированы с диском посредством запроса issue_xlog_fsync
WAL syncs per second	Среднее количество раз, когда файлы WAL синхронизировались с диском с помощью запроса issue_xlog_fsync в секунду
WAL write time (s)	Общее время, затраченное на запись буферов WAL на диск с использованием запроса XLogWrite, в секундах Это включает время синхронизации, когда wal_sync_method является open_datasync или open_sync (wal_write_time)
WAL write duty	Процентное соотношение времени записи WAL от общего времени отчета
WAL sync time (s)	Общее количество времени, затраченного на синхронизацию файлов WAL с диском посредством выполнения запроса issue_xlog_fsync, в секундах
WAL sync duty	Процентное соотношение времени синхронизации WAL от общего времени отчета

4.1.6. Tablespace statistics (Статистика табличных пространств)

Таблица 4.6 содержит информацию о размерах и росте табличных пространств.

Tablespace	Path	Size	Growth
pg_default		37 MB	40 kB
pg_global		559 kB	

Рисунок 4.6 – Пример отчета Tablespace statistics

Таблица 4.6 – Описание параметров отчета Tablespace statistics

Параметр	Описание
Tablespace	имя табличного пространства
Path	путь к табличному пространству
Size	размер табличного пространства на момент последнего снапшота в интервале отчета
Growth	рост табличного пространства в течение интервала отчета

4.2. SQL Query statistics (Статистика SQL-запросов)

В данном разделе отчета содержатся таблицы топ запросов за интервал отчета, отсортированные по нескольким важным статистикам. Данные берутся из представления pg_stat_statements, если оно было доступно во время снятия снапшота.

4.2.1. Top SQL by elapsed time (Топ SQL-запросов по затраченному времени)

Таблица 4.7 содержит топ запросы в количестве, которое указано в параметре `pg_profile.topn`, отсортированные по прошедшему времени `total_plan_time + total_exec_time` из представления `pg_stat_statements`.

`pg_stat_statements.total_plan_time` выключена по умолчанию, для включения этой метрики и расчета данных таблиц необходимо включить параметр `pg_stat_statements.track_planning = 'on'` в `postgresql.conf`.

Query ID	Database	%Total	Time (s)			I/O time (s)		CPU time (s)		Plans	Executions
			Elapsed	Plan	Exec	Read	Write	Usr	Sys		
0adabae509 [6a9719fe696e39f1]	jatoba	97.56	183.58		183.58	0.66					1
e6d419370a [2dd07818c71d61a3]	jatoba	1.68	3.17	0.00	3.17	0.01		2.00	0.17	1	1

Рисунок 4.7 – Пример отчета Top SQL by elapsed time

Таблица 4.7 – Описание параметров отчета Top SQL by elapsed time

Параметр		Описание
Query ID		идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой <code>pgcenter</code> . Родное поле <code>pg_stat_statements queryid</code> в шестнадцатеричной нотации показано в квадратных скобках
Database		имя БД запроса (получено из поля <code>dbid</code>)
%Total		общее время данного утверждения в процентах от общего времени всех утверждений в кластере
Time(s)		время, затраченное на выполнение данного утверждения (в секундах)
	Elapsed	общее время, затраченное на выполнение данного утверждения (<code>total_plan_time + total_exec_time</code>)
	Plan	время, затраченное на планирование этого утверждения (поле <code>total_plan_time</code>)
	Exec	время, затраченное на выполнение данного запроса (поле <code>total_exec_time</code>)
I/O Time(s)		
	Read	время, затраченное на чтение блоков (поле <code>blk_read_time</code>)
	Write	время, затраченное на запись блоков (поле <code>blk_write_time</code>)
CPU time(s)		время, затраченное на работу центрального процессора. Основано на данных, предоставляемых расширением <code>pg_stat_kcache</code>
	Usr	время процессора, проведенное в пространстве пользователя
	Sys	время процессора, затраченное в пространстве ядра
Plans		количество раз, когда утверждение было запланировано (поле <code>plans</code>)
Executions		количество раз, когда оператор был выполнен (поле <code>calls</code>)

4.2.2. Top SQL by planning time (Топ SQL-запросов по времени планирования)

Таблица 4.8 содержит топ запросы в количестве, которое указано в параметре `pg_profile.topn`, отсортированные по полю `total_plan_time` представления `pg_stat_statements`.

Query ID	Database	Plan elapsed (s)	%Elapsed	Plan times (ms)				Plans	Executions
				Mean	Min	Max	StdErr		
2791af6210 [517f788526b8bd80]	jatoba	0.06	66.10	28.374	14.032	42.716	14.342	2	2
415fba67d5 [83539004f15aa819]	jatoba	0.01	13.29	3.408	2.646	4.169	0.762	2	2

Рисунок 4.8 – Пример отчета Top SQL by planning time

Таблица 4.8 – Описание параметров отчета Top SQL by planning time

Параметр		Описание
Query ID		идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой <code>pgcenter</code> . Родное поле <code>pg_stat_statements</code> <code>qieryid</code> в шестнадцатеричной нотации показано в квадратных скобках
Database		имя БД запроса (получено из поля <code>dbid</code>)
Plan elapsed(s)		время, затраченное на планирование данного утверждения (поле <code>total_plan_time</code>)
%Elapsed		время планирования этого утверждения в процентах от времени выполнения утверждения
Plan times (ms)		подробная статистика времени планирования данного утверждения (в миллисекундах)
	Mean	среднее время, затраченное на планирование этого утверждения (поле <code>mean_plan_time</code>)
	Min	минимальное время, затраченное на планирование этого утверждения (поле <code>min_plan_time</code>)
	Max	максимальное время, затраченное на планирование этого утверждения (поле <code>max_plan_time</code>)
	StdErr	популяционное стандартное отклонение времени, затраченное на планирование данного утверждения (поле <code>stddev_plan_time</code>)
Plans		количество раз, когда это утверждение было запланировано (поле <code>plans</code>)
Executions		количество раз, когда этот оператор был выполнен (поле <code>calls</code>)

4.2.3. Top SQL by execution time (Топ SQL-запросов по времени выполнения)

Таблица 4.9 содержит топ запросы в количестве, которое указано в параметре `pg_profile.topn`, отсортированные по полю `total_time` (или `total_exec_time`) представления `pg_stat_statements`.

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

Query ID	Database	Exec (s)	%Elapsed	%Total	I/O time (s)		Rows	Execution times (ms)				Executions
					Read	Write		Mean	Min	Max	StdErr	
e76adc7ebc [b7463840643c9b6b]	jatoba	0.50	99.99	2.88			1	501.053	501.053	501.053		1
6da0f5d137 [6a96292d39ee4a6d]	jatoba	0.24	99.87	1.39	0.00		1	242.366	242.366	242.366		1

Рисунок 4.9 – Пример отчета Top SQL by execution time

Таблица 4.9 – Описание параметров отчета Top SQL by execution time

Параметр		Описание
Query ID		идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой pgcenter. Родное поле pg_stat_statements queryid в шестнадцатеричной нотации показано в квадратных скобках
Database		имя БД запроса (получено из поля dbid)
Exec(s)		время, затраченное на выполнение данного запроса (поле total_exec_time)
%Elapsed		время выполнения этого оператора в процентах от времени выполнения запроса
%Total		время выполнения данного запроса в процентах от общего времени выполнения всех операторов в кластере
I/O time(s):		
	Read	время, затраченное на чтение блоков (поле blk_read_time)
	Write	время, затраченное на запись блоков (поле blk_write_time)
Rows		количество строк, извлеченных или затронутых запросом (поле rows)
Execution times (ms)		подробная статистика времени выполнения данного запроса (в миллисекундах)
	Mean	среднее время выполнения данного запроса (поле mean_exec_time)
	Min	минимальное время, затраченное на выполнение данного запроса (поле min_exec_time)
	Max	максимальное время выполнения данного запроса (поле max_exec_time)
	StdErr	популяционное стандартное отклонение времени, затраченное на выполнение данного запроса (поле stddev_exec_time)
Executions		количество раз, за которое было выполнено данное запроса (поле calls)

4.2.4. Top SQL by executions (Топ SQL-запросов по количеству выполнений)

Таблица 4.10 содержит топ запросы в количестве, которое указано в параметре pg_profile.topn, отсортированные по полю calls представления pg_stat_statements.

Query ID	Database	Executions	%Total	Rows	Mean(ms)	Min(ms)	Max(ms)	StdErr(ms)	Elapsed(s)
1f689ab00c [e0c36159a0d9234b]	jaboba	12	14.81	12	0.018	0.005	0.061	0.019	0.0
1d5d23ec25 [4e671036565abcb7]	jaboba	12	14.81		0.003	0.003	0.005	0.000	0.0

Рисунок 4.10 – Пример отчета Top SQL by executions

Таблица 4.10 – Описание параметров отчета Top SQL by executions

Параметр	Описание
Query ID	идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой pgcenter. Родное поле pg_stat_statements queryid в шестнадцатеричной нотации показано в квадратных скобках
Database	имя БД запроса (получено из поля dbid)
Executions	количество выполнений запроса (поле calls)
%Total	вызовы данного запроса в процентах от общего количества вызовов всех запросов в кластере
Rows	количество строк, извлеченных или затронутых запросом (поле rows)
Mean(ms)	среднее время, проведенное в запросе, в миллисекундах (поле mean_time или mean_exec_time)
Min(ms)	минимальное время, затраченное в запросе, в миллисекундах (поле min_time или min_exec_time)
Max(ms)	максимальное время, проведенное в запросе, в миллисекундах (поле max_time или max_exec_time)
StdErr(ms)	популяционное стандартное отклонение времени, проведенное в запросе, в миллисекундах (поле stddev_time или stddev_exec_time)
Elapsed(s)	количество времени, затраченное на выполнение данного запроса, в секундах (поле total_time или total_exec_time)

4.2.5. Top SQL by I/O wait time (Топ SQL-запросов по времени ожидания ввода/вывода)

Таблица 4.11 содержит топ запросы в количестве, которое указано в параметре pg_profile.topn, отсортированные по времени чтения и записи (blk_read_time + blk_write_time).

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

Query ID	Database	IO(s)	R(s)	W(s)	%Total	Reads			Writes			Elapsed(s)	Executions
						Shr	Loc	Tmp	Shr	Loc	Tmp		
0adabae509 [6a9719fe696e39f1]	jatoba	0.210	0.210		47.28	2703	500		6800	16000		101.6	1
af486fb8c9 [a3e0abc92178bf5]	jatoba	0.184	0.184		41.55	487			7			0.7	1

Рисунок 4.11 – Пример отчета Top SQL by I/O wait time

Таблица 4.11 – Описание параметров отчета Top SQL by I/O wait time

Параметр		Описание
Query ID		идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой pgcenter. Родное поле pg_stat_statements queryid в шестнадцатеричной нотации показано в квадратных скобках
Database		имя БД запроса (получено из поля dbid)
IO(s)		количество времени, затраченное на чтение и запись (время ввода/вывода) данным утверждением в секундах (blk_read_time + blk_write_time)
R(s)		количество времени, затраченное на чтение, данным оператором в секундах (blk_read_time)
W(s)		количество времени, затраченное на запись, данным оператором в секундах (blk_write_time)
%Total		время ввода/вывода данного оператора в процентах от общего времени ввода/вывода для всех операторов в кластере
Reads		количество блоков, прочитанных данным оператором, разделенное на три подстолбца
	Shr	общие чтения (поле shared_blks_read)
	Loc	локальные чтения (поле local_blks_read)
	Tmp	временные чтения (поле temp_blks_read)
Writes		количество блоков, записанных данным оператором, разделенное на три подстолбца
	Shr	общие записи (поле shared_blks_written)
	Loc	локальные записи (поле local_blks_written)
	Tmp	временные записи (поле temp_blks_written)
Elapsed(s)		количество времени, затраченное на выполнение данного запроса в секундах (поле total_time или total_exec_time)
Executions		количество выполнений для данного запроса (поле calls)

4.2.6. Top SQL by shared blocks fetched (Топ SQL-запросов по выбранным общим блокам)

Таблица 4.12 содержит топ запросы в количестве, которое указано в параметре pg_profile.topn, отсортированные по прочитанным и перехваченным блокам, помогающие обнаружить запросы которые обрабатывают наибольшее количество данных.

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

Query ID	Database	blks fetched	%Total	Hits(%)	Elapsed(s)	Rows	Executions
0577c86d25 [98f4aaa0e6ed0b94]	jaboba	1382515	97.24	100.00	1.2	2	2
d19c3556fe [1e65f960766458a1]	jaboba	17638	1.24	96.98	0.3	1	1

Рисунок 4.12 – Пример отчета Top SQL by shared blocks fetched

Таблица 4.12 – Описание параметров отчета Top SQL by shared blocks fetched

Параметр	Описание
Query ID	идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой pgcenter. Родное поле pg_stat_statements queryid в шестнадцатеричной нотации показано в квадратных скобках
Database	имя БД запроса (получено из поля dbid)
blks fetched	количество найденных блоков (выражение shared_blks_hit + shared_blks_read)
%Total	количество блоков, найденных для данного запроса в процентах от общего количества блоков, найденных для всех утверждений в кластере
Hits(%)	процент блоков, полученных из буферов в пределах всех полученных блоков
Elapsed(s)	количество времени, затраченное на выполнение данного запроса, в секундах (поле total_time или total_exec_time+total_plan_time)
Rows	количество строк, извлеченных или затронутых данным запросом (поле rows)
Executions	количество выполнений для данного запроса (поле calls)

4.2.7. Top SQL by shared blocks read (Топ SQL-запросов по количеству прочитанных разделяемых блоков)

Таблица 4.13 содержит топ запросы в количестве, которое указано в параметре pg_profile.topn, отсортированные по общим прочитанным блокам, помогающие обнаружить наиболее интенсивные по чтению операторы.

Query ID	Database	Reads	%Total	Hits(%)	Elapsed(s)	Rows	Executions
b251f96891 [b4079dbd7c7c49dc]	jaboba	649	46.03	85.03	0.1		1
d19c3556fe [1e65f960766458a1]	jaboba	532	37.73	96.98	0.3	1	1

Рисунок 4.13 – Пример отчета Top SQL by shared blocks read

Таблица 4.13 – Описание параметров отчета Top SQL by shared blocks read

Параметр	Описание
Query ID	идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой pgcenter. Родное поле pg_stat_statements queryid в шестнадцатеричной нотации показано в квадратных скобках
Database	имя БД запроса (получено из поля dbid)
Reads	количество общих блоков чтения для данного запроса (поле shared_blks_read)

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

Параметр	Описание
%Total	количество общих чтений для данного запроса в процентах от общего количества общих чтений для всех утверждений в кластере
Hits(%)	процент блоков, полученных из буферов, среди всех полученных блоков
Elapsed(s)	количество времени, затраченное на выполнение данного запроса в секундах (поле total_time или total_exec_time+total_plan_time)
Rows	количество строк, извлеченных или затронутых данным запросом (поле rows)
Executions	количество выполнений для данного запроса (поле calls)

4.2.8. Top SQL by shared blocks dirtied (Топ SQL-запросов по заполненным разделяемым блокам)

Таблица 4.14 содержит топ запросы в количестве, которое указано в параметре pg_profile.topn, отсортированные по количеству общих заполненных буферов, помогающие обнаружить наиболее изменяющие данные запросы.

Query ID	Database	User	Dirtied	%Total	Hits(%)	Dirtied	WAL	%Total	Elapsed(s)	Rows	Executions
947e7563da889e91 [d071cc622a]	pg_profile	postgres	248	248	100	99.2	1629 kB	82.79	2.04	1	1
51aa50e95c6051fd [d203973f9b]	postgres	postgres	1	1	99.9	0.4			0.04	109	1
51aa50e95c6051fd [dfdd6c545f]	pg_profile	postgres	1	1	99.94	0.4			0.03	183	1

Рисунок 4.14 – Пример отчета Top SQL by shared blocks dirtied

Таблица 4.14 – Описание параметров отчета Top SQL by shared blocks dirtied

Параметр	Описание
Query ID	идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой pgcenter. Родное поле pg_stat_statements queryid в шестнадцатеричной нотации показано в квадратных скобках
Database	имя БД запроса (получено из поля dbid)
User	Имя пользователя
Dirtied	количество общих блоков, заполненных этим запросом (поле shared_blks_dirtied)
%Total	количество общих блоков, заполненных этим запросом, в процентах от общего количества общих блоков, заполненными всеми операторами в кластере
Hits(%)	процент блоков, полученных из буферов, среди всех полученных блоков
Elapsed(s)	количество времени, затраченное на выполнение данного запроса в секундах (поле total_time или total_exec_time+total_plan_time)
Rows	количество строк, извлеченных или затронутых данным запросом (поле rows)
Executions	количество выполнений для данного запроса (поле calls)

4.2.9. Top SQL by shared blocks written (Топ SQL-запросов по записи общих блоков)

Таблица 4.15 содержит топ запросы в количестве, которое указано в параметре `pg_profile.topn`, выполняющие записи отсортированные по количеству записанных блоков.

Query ID	Database	User	Written	%Total	%BackendW	Hits(%)	Elapsed(s)	Rows	Executions
947e7563da889e91 [d071cc622a]	pg_profile	postgres	73			100	2.04	1	1

Рисунок 4.15 – Пример отчета Top SQL by shared blocks written

Таблица 4.15 – Описание параметров отчета Top SQL by shared blocks written

Параметр	Описание
Query ID	идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой <code>pgcenter</code> . Родное поле <code>pg_stat_statements queryid</code> в шестнадцатеричной нотации показано в квадратных скобках
Database	имя БД запроса (получено из поля <code>dbid</code>)
User	Имя пользователя
Written	количество блоков, записанных данным запросом (поле <code>shared_blks_written</code>)
%Total	количество блоков, записанных данным запросом в процентах от общего количества блоков, записанных всеми операторами в кластере
%BackendW	количество блоков, записанных данным запросом, в процентах от всех блоков, записанных backend в кластере (поле <code>buffers_backend</code> представления <code>pg_stat_bgwriter</code>)
Hits(%)	процент блоков, полученных из буферов, среди всех полученных блоков
Elapsed(s)	количество времени, затраченное на выполнение данного запроса в секундах (поле <code>total_time</code> или <code>total_exec_time+total_plan_time</code>)
Rows	количество строк, извлеченных или затронутых данным запросом (поле <code>rows</code>)
Executions	количество выполнений данного запроса (поле <code>calls</code>)

4.2.10. Top SQL by WAL size (Топ SQL-запросов по размеру WAL)

Таблица 4.16 содержит топ запросы в количестве, которое указано в параметре `pg_profile.topn`, отсортированные по размеру WAL (доступно в `pg_stat_statements`, начиная с версии 1.8).

Query ID	Database	User	WAL	%Total	Dirtied	WAL FPI	WAL records
947e7563da889e91 [d071cc622a]	pg_profile	postgres	1629 kB	82.79	248	159	6727
eb976b7a5390fb35 [f133a6ab3a]	pg_profile	postgres	168 bytes	0.01			2

Рисунок 4.16 – Пример отчета Top SQL by WAL size

Таблица 4.16 – Описание параметров отчета Top SQL by WAL size

Параметр	Описание
Query ID	идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой pgcenter. Родное поле pg_stat_statements queryid в шестнадцатеричной нотации показано в квадратных скобках
Database	имя БД запроса (получено из поля dbid)
User	Имя пользователя
WAL	объем WAL, сгенерированный запросом (поле wal_bytes)
%Total	объем WAL, сгенерированный оператором, как процент от общего объема WAL, сгенерированного в кластере (приращение pg_current_wal_lsn())
Dirtied	количество общих блоков, заполненных этим запросом (поле shared_blks_dirtied)
WAL FPI	общее количество полностраничных образов WAL, сгенерированных этим запросом (поле wal_fpi)
WAL records	общее количество байт WAL, сгенерированных запросом (поле wal_bytes)

4.2.11. Complete list of SQL texts (Полный список текстов SQL)

Таблица содержит полный текст SQL-запросов и состоит из столбцов:

- Query ID – идентификатор запроса в виде хэша БД;
- Query Text – текст SQL-запроса.

Complete list of SQL texts

Query ID	
844260166fd58c94	SELECT st.*,stio.idx_blks_read,stio.idx_blks_hit,CASE l.relation WHEN st.indexrelid THEN \$1 l (ix.indexrelid = st.indexrelid) JOIN pg_catalog.pg_class ON (pg_class.oid = st.indexrelid) LI
8dba5df162075f82	SET application_name='pg_profile'
919b6da4fddfe820	SELECT dbs.datid, dbs.datname, dbs.xact_commit, dbs.xact_rollback, dbs.blks_read, dbs.blks_h: as datsize, \$1 as datsize_delta, db.datistemplate, db.dattablespace, db.dataallowconn FROM pg
947e7563da889e91	SELECT profile.take_sample(\$1, \$2)
9b4306b40963f430	SELECT name, reset_val, unit, pending_restart FROM pg_catalog.pg_settings WHERE name IN (\$1)
ab9c2616533b2346	SET search_path=''
c5b7234c6643e938	DISCARD ALL
d02bac127fc7aa33	SET lock_timeout=3000
d4de9d554b22c942	SELECT extname, extnamespace::regnamespace::name AS extnamespace, extversion FROM pg_catalog
eb976b7a5390fb35	SELECT profile.get_report(\$1, \$2, \$3, \$5, \$4) AS "Html"
f09fe0570da9db74	SELECT oid as tablespaceid,spcname as tablespacename,pg_catalog.pg_tablespace_location(oid) :
b979761ebaa9492	SELECT system_identifier FROM pg_catalog.pg_control_system()

Рисунок 4.17 - Пример отчета Complete list of SQL texts

4.2.12. Top SQL by temp usage (Топ SQL-запросов по использованию временных данных)

Таблица 4.18 содержит топ запросы в количестве, которое указано в параметре pg_profile.topn, отсортированные по temp I/O, рассчитанные как сумма полей temp_blks_read, temp_blks_written, local_blks_read и local_blks_written.

Query ID	Database	Local fetched	Hits(%)	Local (blk)				Temp (blk)				Elapsed(s)	Rows	Executions
				Write	%Total	Read	%Total	Write	%Total	Read	%Total			
0577c86d25 [98f4aaa9e6ed0b94]	jaboba	366	99.45	4	100.00	2	100.00					1.2	2	2

Рисунок 4.18 – Пример отчета Top SQL by temp usage

Таблица 4.17 – Описание параметров отчета Top SQL by temp usage

Параметр	Описание
Query ID	идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой pgcenter. Родное поле pg_stat_statements queryid в шестнадцатеричной нотации показано в квадратных скобках
Database	имя БД запроса (получено из поля dbid)
Local fetched	количество извлеченных локальных блоков (выражение local_blks_hit + local_blks_read)
Hits(%)	процент локальных блоков, полученных из временных буферов, от всех полученных локальных блоков
Local (blk)	статистика ввода/вывода блоков, используемых во временных таблицах
Write	количество записанных локальных блоков (local_blks_written)

Параметр		Описание
	%Total	количество локальных блоков, записанных данным запросом в процентах от общего количества локальных блоков, записанных для всех операторов в кластере
	Read	количество прочитанных локальных блоков (local_blks_read)
	%Total	количество локальных блоков, прочитанных данным запросом в процентах от общего количества локальных блоков, прочитанных всеми операторами в кластере
Temp (blk)		статистика ввода/вывода блоков, используемых в операциях сортировки и объединения
	Write	количество записанных временных блоков (temp_blks_written)
	%Total	количество записанных временных блоков данным запросом в процентах от общего количества записанных временных блоков для всех операторов в кластере
	Read	количество прочитанных локальных блоков (temp_blks_read)
	%Total	количество прочитанных локальных блоков данным запросом в процентах от общего количества прочитанных локальных блоков для всех операторов в кластере
Elapsed(s)		количество времени, затраченного на выполнение данного запроса в секундах (поле total_time или total_exec_time+total_plan_time)
Rows		количество строк, извлеченных или затронутых данным запросом (поле rows)
Executions		количество выполнений для данного запроса (поле calls)

4.3. Rusage statistics (Статистика использования ресурсов)

Данный раздел содержит статистику использования ресурсов, предоставленную расширением pg_stat_kcache, если оно было доступно в интервале отчета.

4.3.1. Top SQL by system and user time (Топ SQL-запросов по потреблению системного и пользовательского времени)

Таблица 4.19 содержит топ запросы в количестве, которое указано в параметре pg_profile.topn, отсортированные по сумме полей user_time и system_time расширения pg_stat_kcache.

Query ID	Database	User Time			System Time		
		Plan (s)	Exec (s)	%Total	Plan (s)	Exec (s)	%Total
b8bfd7e5fd [d95f0d5c48ef20ec]	jatoba	0.00	0.60	0.78	0.00	0.45	4.15
af486fb8c9 [a3e0abc92178bf5]	jatoba	0.00	0.18	0.23		0.14	1.30

Рисунок 4.19 – Пример отчета Top SQL by system and user time

Таблица 4.18 – Описание параметров отчета Top SQL by system and user time

Параметр		Описание
Query ID		идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой pgcenter. Родное поле pg_stat_statements queryid в шестнадцатеричной нотации показано в квадратных скобках
Database		имя БД запроса (получено из поля dbid)
User Time		использованное процессорное время пользователя
	Plan (s)	время процессора пользователя, затраченное на планирование в секундах (поле plan_user_time)
	Exec (s)	процессорное время пользователя, затраченное на выполнение в секундах (поле exec_user_time или user_time)
	%Total	пользовательское процессорное время данного запроса в процентах от суммарного пользовательского процессорного времени для всех утверждений
System Time		использованное системное процессорное время
	Plan (s)	системное процессорное время, прошедшее во время планирования в секундах (поле plan_system_time)
	Exec (s)	системное процессорное время, затраченное на выполнение в секундах (поле exec_system_time или system_time)
	%Total	системное процессорное время данного запроса в процентах от суммарного системного процессорного времени для всех операторов

4.3.2. Top SQL by reads/writes done by filesystem layer (Топ SQL-запросов по количеству операций чтения/записи, выполняемых на уровне файловой системы)

Таблица 4.20 содержит топ запросы в количестве, которое указано в параметре pg_profile.topn, отсортированные по сумме полей reads и writes поля pg_stat_kcache.

Query ID	Database	Read Bytes			Write Bytes		
		Plan	Exec	%Total	Plan	Exec	%Total
af486fb8c9 [a3e0abc92178bf5]	jatoba		1744 kB	28.68		56 kB	0.03
1f404a0f19 [1ddcf6c3e481473d]	jatoba		284 kB	4.67			

Рисунок 4.20 – Пример отчета Top SQL by reads/writes done by filesystem layer

Таблица 4.19 – Описание параметров отчета Top SQL by reads/writes done by filesystem layer

Параметр		Описание
Query ID		идентификатор запроса в виде хэша БД, пользователя и текста запроса. Совместим с утилитой pgcenter. Родное поле pg_stat_statements queryid в шестнадцатеричной нотации показано в квадратных скобках
Database		имя БД запроса (получено из поля dbid)
Read Bytes		количество байт, прочитанных уровнем файловой системы
	Plan	байты, прочитанные во время планирования (поле plan_reads)
	Exec	байты, прочитанные во время выполнения (поле exec_reads)
	%Total	байты, прочитанные данным запросом как процент от суммарного количества прочитанных байтов для всех операторов
Write Bytes		количество байт, записанных уровнем файловой системы (поле writes)
	Plan	байты, записанные во время планирования (поле plan_writes)
	Exec	байты, записанные во время выполнения (поле exec_writes)
	%Total	байты, записанные данным запросом как процент от общего количества байт записанных всеми операторами

4.3.3. Top SQL by JIT elapsed time (Топ SQL-запросов отсортированных по общему JIT-времени)

Таблица «Top SQL by JIT elapsed time» Топ SQL-запросов, отсортированных по суммарному JIT-времени, связанному со следующими параметрами:

- jit_generation_time;
- jit_inlining_time;
- jit_optimization_time;
- jit_emission_time.

Таблица 4.20 – Описание параметров отчета Top SQL by JIT elapsed time

Параметр		Описание
Query ID		идентификатор запроса, предоставленный расширением pg_stat_statements (queryid), в шестнадцатеричной нотации. Альтернативный идентификатор запроса в виде хэша из dbid, userid и queryid показан в квадратных скобках, этот идентификатор совместим с утилитой pgcenter.
Database		имя базы данных, в которой было выполнено заявление (получено из поля dbid)
User		имя пользователя, выполнившего это заявление (получено из field userid)
JIT total (s)		время, затрачиваемое на генерацию JIT в секундах (jit_generation_time + jit_inlining_time + jit_optimization_time + jit_emission_time)
Generation		генерация
	Count	общее количество функций, скомпилированных JIT для заявления
	Gen. time	общее время, которое заявление потратило на генерацию кода JIT
Inlining		Встраивание
	Count	количество раз, когда функции были встроены
	Time	общее количество времени, затраченного заявлением на встраивание функций (в секундах)
Optimization		оптимизация
	Count	количество раз, сколько заявление было оптимизировано
	Time	общее количество секунд, затраченных заявлением на оптимизацию
Emission		эмиссия
	Count	количество раз, сколько был сгенерирован код
	Time	общее потраченное время заявлением на сгенерированном коде в секундах
Time (s)		время, затраченное этим заявлением в секундах
	Plan	время, затраченное планом на планирование этого утверждения (field total_plan_time в поле)
	Exec	Время, затраченное выполнением этого запроса (поле total_exec_time).
I/O time (s)		Время ввода-вывода (с)
	Read	Время, затраченное на прочтение блоков (field blk_read_time).
	Write	Время, затраченное на запись блоков (поле blk_write_time).

4.3.4. Complete list of SQL texts (Полный список текстов SQL-запросов)

Таблица 4.21 содержит тексты всех запросов, упомянутых в отчете. Можно использовать ссылку Query ID в любой таблице статистики, чтобы перейти туда и увидеть текст запроса.

QueryID	Query Text
7430f61983	SET search_path=''
1998b65ef0	select * from pg_stat_statements pss

Рисунок 4.21 – Пример отчета Complete list of SQL texts

4.4. Schema object statistics (Статистика объекта схемы)

Данный раздел отчета содержит топ объектов БД, используя статистику из представлений Statistics Collector.

4.4.1. Top tables by estimated sequentially scanned volume (Топ таблиц по предполагаемому объему последовательного сканирования)

Таблица 4.22 содержит топ запросы БД, отсортированные по предполагаемому объему, прочитанному последовательным сканированием. Основано на представлении pg_stat_all_tables. Здесь можно искать таблицы, в которых, возможно, отсутствует какой-либо индекс.

DB	Tablespace	Schema	Table	~SeqBytes	SeqScan	IxScan	IxFet	Ins	Upd	Del	Upd(HOT)
jatoba	pg_default	profile	tablespaces_list	19 GB	411004	2	2				
			tablespaces_list(TOAST)								

Рисунок 4.22 – Пример отчета Top tables by estimated sequentially scanned volume

Таблица 4.21 – Описание параметров отчета Top tables by estimated sequentially scanned volume

Параметр	Описание
Database	имя БД, в которой находится таблица
Tablespace	имя табличного пространства, в котором находится таблица
Schema	имя схемы таблицы
Table	имя таблицы
~SeqBytes	предполагаемый объем, прочитанный при последовательном сканировании. Рассчитывается как сумма размера отношения, умноженная на seq_scan для всех снапшотов отчета
SeqScan	количество последовательных сканирований, выполненных над таблицей (поле seq_scan)
IxScan	количество индексных сканирований, инициированных для этой таблицы (поле idx_scan)
IxFet	количество живых строк, извлеченных при индексном сканировании (поле idx_tup_fetch)
Ins	количество вставленных строк (поле n_tup_ins)
Upd	количество обновленных строк (включая HOT) (поле n_tup_upd)
Del	количество удаленных строк (поле n_tup_del)

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

Параметр	Описание
Upd(HOT)	количество строк, обновленных по HOT (поле n_tup_hot_upd)

4.4.2. Top tables by blocks fetched (Топ таблиц по выбранным блокам)

Таблица 4.23 содержит захваченный блок – это блок, обрабатываемый с диска (read) или из общих буферов (hit). Таблицы в этом списке сортируются по сумме найденных блоков для отношения таблицы, ее индексов, TOAST-таблицы (если существует) и индекса TOAST (если существует). Этот раздел может привлечь внимание к таблицам с избыточной обработкой блоков. Основано на данных представления pg_statio_all_tables.

DB	Tablespace	Schema	Table	Heap		Ix		TOAST		TOAST-Ix	
				Blks	%Total	Blks	%Total	Blks	%Total	Blks	%Total
jatoba	pg_default	profile	sample_stat_tables	1406487	16.06	964091	11.01				
jatoba	pg_default	profile	tables_list	598622	6.83	829703	9.47				

Рисунок 4.23 – Пример отчета Top tables by blocks fetched

Таблица 4.22 – Описание параметров отчета Top tables by blocks fetched

Параметр	Описание
Database	имя БД таблицы
Tablespace	имя табличного пространства, в котором находится таблица
Schema	имя схемы таблицы
Table	имя таблицы
Heap	статистика по найденным блокам таблиц (heap_blks_read + heap_blks_hit)
Ix	статистика для всех найденных блоков индексов таблиц (idx_blks_read + idx_blks_hit)
TOAST	статистика для блоков, полученных для TOAST-таблиц (toast_blks_read + toast_blks_hit)
TOAST-Ix	статистика для найденных блоков индекса TOAST (tidx_blks_read + tidx_blks_hit)
Каждое поле статистики в этой таблице разделено на два столбца	
Blks	количество блоков, найденных для heap таблиц, индекса, TOAST или индекса TOAST
%Total	блоки, найденные для heap таблиц, индекса, TOAST или индекса TOAST в процентах от всех блоков, найденных во всем кластере

4.4.3. Top tables by blocks read (Топ таблиц по прочитанным блокам)

Таблица 4.24 содержит топ таблицы, отсортированные по количеству прочитанных блоков. Таблицы в этом списке отсортированы по сумме прочитанных блоков для таблицы, ее индексов, TOAST-таблицы (если существует) и индекса TOAST (если существует). Этот раздел может сфокусировать внимание на таблицах с чрезмерным количеством прочитанных блоков. Основано на данных представления pg_statio_all_tables.

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

DB	Tablespace	Schema	Table	Heap		Ix		TOAST		TOAST-Ix		Hit(%)
				Blks	%Total	Blks	%Total	Blks	%Total	Blks	%Total	
jatoba	pg_default	pg_catalog	pg_attribute	2280	14.31	3404	21.37					99.51
jatoba	pg_default	pg_catalog	pg_statistic	3139	19.70	575	3.61	6	0.04	2	0.01	99.58

Рисунок 4.24 – Пример отчета Top tables by blocks read

Таблица 4.23 – Описание параметров отчета Top tables by blocks read

Параметр	Описание
Database	имя БД таблицы
Tablespace	имя табличного пространства, в котором находится таблица
Schema	имя схемы таблицы
Table	имя таблицы
Heap	статистика чтения блоков таблиц (heap_blks_read)
Ix	статистика чтения блоков всех индексов таблицы (idx_blks_read)
TOAST	статистика чтений блоков TOAST-таблиц (toast_blks_read)
TOAST-Ix	статистика чтения блоков индексов TOAST (tidx_blks_read)
Hit(%)	процент блоков, полученных из буферов, среди всех полученных блоков
Каждая статистика чтения в этой таблице разделена на два столбца	
Blks	количество блочных чтений для heap таблиц, index, TOAST или индекса TOAST
%Total	блочные чтения для heap таблиц, индекса, TOAST или индекса TOAST как процент от всех блочных чтений во всем кластере

4.4.4. Top DML tables (Топ таблиц по количеству операций DML)

Таблица 4.25 содержит топ таблицы, отсортированные по количеству строк, затронутых DML, т.е. сумма n_tup_ins, n_tup_upd и n_tup_del (включая таблицы TOAST).

DB	Tablespace	Schema	Table	Ins	Upd	Del	Upd(HOT)	SeqScan	SeqFet	IxScan	IxFet
jatoba	pg_default	pg_catalog	pg_attribute	140512		140500		33	156	131780	531396
jatoba	pg_default	pg_catalog	pg_statistic	97500	315	97500	186			182467	176685
			pg_statistic(TOAST)	3		2				15	14

Рисунок 4.25 – Пример отчета Top DML tables

Таблица 4.24 – Описание параметров отчета Top DML tables

Параметр	Описание
Database	имя БД таблиц
Tablespace	имя табличного пространства, в котором находится таблица
Schema	имя схемы таблицы
Table	имя таблицы
Ins	количество вставленных строк (поле n_tup_ins)
Upd	количество обновленных строк (включая HOT) (поле n_tup_upd)
Del	количество удаленных строк (поле n_tup_del)
Upd(HOT)	количество обновленных строк HOT (поле n_tup_hot_upd)

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

Параметр	Описание
SeqScan	количество последовательных сканирований, выполненных над таблицей (поле seq_scan)
SeqFet	количество живых строк, извлеченных в результате последовательного сканирования (поле seq_tup_read)
IxScan	количество индексных сканирований, инициированных для этой таблицы (поле idx_scan)
IxFet	количество живых строк, полученных при индексном сканировании (поле idx_tup_fetch)

4.4.5. Top tables by updated/deleted tuples (Топ таблиц по обновленным/удаленным записям)

Таблица 4.26 содержит топ таблицы, отсортированные по количеству операций, вызывающих автовакуумную нагрузку, т.е. сумма n_tup_upd и n_tup_del (включая таблицы TOAST).

DB	Tablespace	Schema	Table	Upd	Upd(HOT)	Del	Vacuum	AutoVacuum	Analyze	AutoAnalyze
jatoba	pg_default	pg_catalog	pg_attribute			140500		2		1
jatoba	pg_default	pg_catalog	pg_statistic	315	186	97500		2		
			pg_statistic(TOAST)			2				

Рисунок 4.26 – Пример отчета Top tables by updated/deleted tuples

Таблица 4.25 – Описание параметров отчета Top tables by updated/deleted tuples

Параметр	Описание
Database	имя БД таблицы
Tablespace	имя табличного пространства, в котором находится таблица
Schema	имя схемы таблицы
Table	имя таблицы
Upd	количество обновленных строк (включая HOT) (поле n_tup_upd)
Upd(HOT)	количество строк, обновленных HOT (поле n_tup_hot_upd)
Del	количество удаленных строк (поле n_tup_del)
Vacuum	количество раз, когда эта таблица была обработана вручную (не считая VACUUM FULL) (поле vacuum_count)
AutoVacuum	количество раз, когда эта таблица была обработана фоновым процессом autovacuum (поле autovacuum_count)
Analyze	количество раз, когда эта таблица была проанализирована вручную (поле analyze_count)
AutoAnalyze	количество раз, когда эта таблица была проанализирована процессом autovacuum (поле autoanalyze_count)

4.4.6. Top growing tables (Топ таблиц по увеличению размера)

Таблица 4.27 содержит топ таблицы, отсортированные по росту.

DB	Tablespace	Schema	Table	Size	Growth	Ins	Upd	Del	Upd(HOT)
pg_profile	pg_default	pg_catalog	pg_statistic	512 kB	48 kB	4	424		80
			pg_statistic(TOAST)	80 kB	8192 bytes	6		3	
pg_profile	pg_default	profile	funcs_list	48 kB	32 kB	27	2		
			funcs_list(TOAST)						

Рисунок 4.27 – Пример отчета Top growing tables

Таблица 4.26 – Описание параметров отчета Top growing tables

Параметр	Описание
Database	имя БД таблицы
Tablespace	имя табличного пространства, в котором находится таблица
Schema	имя схемы таблицы
Table	имя таблицы
Size	размер таблицы, каким он был на момент последнего снапшота в интервале отчета
Growth	рост таблицы
Ins	количество вставленных строк (поле n_tup_ins)
Upd	количество обновленных строк (включая HOT) (поле n_tup_upd)
Del	количество удаленных строк (поле n_tup_del)
Upd(HOT)	количество строк, обновленных по HOT (поле n_tup_hot_upd)

4.4.7. Top indexes by blocks fetched (Топ индексов по выбранным блокам)

Таблица 4.28 содержит захваченный блок – это блок, обрабатываемый с диска (read) или из общих буферов (hit). Основано на данных представления pg_statio_all_indexes.

DB	Tablespace	Schema	Table	Index	Scans	Blks	%Total
pg_profile	pg_default	profile	last_stat_tables_srv1	pk_last_stat_tables_srv1	8358	24943	1.48
pg_profile	pg_default	pg_catalog	pg_class	pg_class_oid_index	3001	6005	0.36
pg_profile	pg_default	profile	last_stat_indexes_srv1	pk_last_stat_indexes_srv1	1076	4944	0.29

Рисунок 4.28 – Пример отчета Top indexes by blocks fetched

Таблица 4.27 – Описание параметров отчета Top indexes by blocks fetched

Параметр	Описание
Database	имя БД индекса
Tablespace	имя табличного пространства, в котором расположен индекс
Schema	имя схемы индекса
Table	имя таблицы
Index	имя индекса

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

Параметр	Описание
Scans	количество сканирований, выполненных по индексу (поле idx_scan)
Blks	блоки, извлеченные из данного индекса (idx_blks_read + idx_blks_hit)
%Total	блоки, считанные для этого индекса, в процентах от всех блоков, считанных во всем кластере

4.4.8. Top indexes by blocks read (Топ индексов по прочитанным блокам)

Таблица 4.29 содержит топ индексы, отсортированные по количеству прочитанных блоков. Основано на данных представления pg_statio_all_indexes.

DB	Tablespace	Schema	Table	Index	Scans	Blks Reads	%Total	Hits(%)
pg_profile	pg_default	pg_catalog	pg_statistic	pg_statistic_relid_att_inh_index	860	2	2.25	99.92
pg_profile	pg_default	profile	sample_stat_tables	pk_sample_stat_tables		2	2.25	98.91
pg_profile	pg_default	profile	last_stat_tables_srv1	pk_last_stat_tables_srv1	8358	2	2.25	99.99

Рисунок 4.29 – Пример отчета Top indexes by blocks read

Таблица 4.28 – Описание параметров отчета Top indexes by blocks read

Параметр	Описание
Database	имя БД индекса
Tablespace	имя табличного пространства, в котором расположен индекс
Schema	имя схемы индекса
Table	имя таблицы
Index	имя индекса
Scans	количество сканирований, выполненных по индексу (поле idx_scan)
Blk Reads	количество дисковых блоков, прочитанных из данного индекса (поле idx_blks_read)
%Total	чтение блоков из данного индекса в процентах от всех чтений блоков во всем кластере
Hits(%)	процент индексных блоков, полученных из буферного кэша, среди всех индексных блоков, найденных для этого индекса

4.4.9. Top growing indexes (Топ таблиц по увеличению объемов индексов)

Таблица 4.30 содержит топ индексы, отсортированные по росту.

DB	Tablespace	Schema	Table	Index	Index		Table		
					Size	Growth	Ins	Upd	Del
jatoba	pg_default	pg_catalog	pg_depend	pg_depend_depender_index	1560 kB	128 kB	18507		18500
jatoba	pg_default	pg_catalog	pg_depend	pg_depend_reference_index	1656 kB	72 kB	18507		18500
jatoba	pg_default	profile	html_reports(TOAST)	pg_toast_17254_index	472 kB	56 kB	9500		

Рисунок 4.30 – Пример отчета Top growing indexes

Таблица 4.29 – Описание параметров отчета Top growing indexes

Параметр	Описание
Database	имя БД индекса
№ изменения: _____	
Подпись отв. лица: _____	
Дата внесения изм: _____	

Параметр		Описание
Tablespace		имя табличного пространства, в котором находится индекс
Schema		имя схемы индекса
Table		имя таблицы
Index		имя индекса
Index		статистика индекса
	Size	размер индекса, каким он был на момент последнего снимка в интервале отчета
	Growth	рост индекса в течение интервала отчета
Table		статистика базовой таблицы
	Ins	количество вставленных строк в базовую таблицу (поле n_tup_ins)
	Upd	количество строк, обновленных в базовой таблице (без HOT) (n_tup_upd - n_tup_hot_upd)
	Del	количество строк, удаленных из базовой таблицы (поле n_tup_del)

4.4.10. Unused indexes (Неиспользуемые индексы)

Таблица 4.31 содержит неиспользуемые индексы в течение интервала отчета, отсортированные по операциям DML над базовыми таблицами, вызывающими поддержку индексов. Индексы ограничений исключены.

DB	Tablespaces	Schema	Table	Index	Index		Table		
					Size	Growth	Ins	Upd	Del
jatoba	pg_default	pg_catalog	pg_class	pg_class_tblspc_relfilenode_index	376 kB	32 kB	7502	11	7500
jatoba	pg_default	profile	sample_statements	ix_sample_stmts_qid	16 kB		38		

Рисунок 4.31 – Пример отчета Unused indexes

Таблица 4.30 – Описание параметров отчета Unused indexes

Параметр		Описание
Database		имя БД индекса
Tablespace		имя табличного пространства, в котором расположен индекс
Schema		имя схемы индекса
Table		имя таблицы
Index		имя индекса
Index		статистика индекса
	Size	размер индекса, каким он был на момент последнего снимка в интервале отчета
	Growth	рост индекса в течение интервала отчета
Table		статистика базовой таблицы
	Ins	количество вставленных строк в базовую таблицу (поле n_tup_ins)
	Upd	количество строк, обновленных в базовой таблице (без HOT) (n_tup_upd - n_tup_hot_upd)
	Del	количество строк, удаленных из базовой таблицы (поле n_tup_del)

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

4.5. User function statistics (Статистика функций пользователя)

Данный раздел отчета содержит топ функции в кластере, основанные на представлении pg_stat_user_functions, которое наполняется данными о статистике выполнения функций.

4.5.1. Top functions by total time (Топ функций по общему времени)

Таблица 4.32 содержит топ функций, отсортированных по затраченному времени.

DB	Schema	Function	Executions	Time (s)			
				Total	Self	Mean	Mean self
jatoba	profile	get_report	500	95.13	0.18	0.190	0.000
jatoba	profile	get_report	500	94.89	20.70	0.190	0.041
jatoba	profile	top_indexes	500	15.87	15.87	0.032	0.032

Рисунок 4.32 – Пример отчета Top functions by total time

Таблица 4.31 – Описание параметров отчета Top functions by total time

Параметр		Описание
Database		имя БД функции
Schema		имя схемы индекса
Function		имя функции
Executions		количество вызовов данной функции (поле calls)
Time (s)		статистика времени выполнения функции в секундах
	Total	общее время, затраченное на эту функцию и все другие вызванные ей функции (поле total_time)
	Self	общее время, затраченное на выполнение самой функции без учета других вызванных ей функций (поле self_time)
	Mean	среднее время выполнения одной функции
	Mean self	среднее собственное время выполнения одной функции

4.5.2. Top functions by executions (Топ функций по исполнению)

Таблица 4.33 содержит топ функции, отсортированные по количеству выполнений.

DB	Schema	Function	Executions	Time (s)			
				Total	Self	Mean	Mean self
jatoba	profile	collect_queries	55500	5.01	5.01	0.000	0.000
jatoba	profile	jsonb_replace	18000	6.08	6.08	0.000	0.000

Рисунок 4.33 – Пример отчета Top functions by executions

Таблица 4.32 – Описание параметров отчета Top functions by executions

Параметр		Описание
Database		имя БД функции
№ изменения: _____		Подпись отв. лица: _____ Дата внесения изм: _____

Параметр		Описание
Schema		имя схемы индекса
Function		имя функции
Executions		количество вызовов данной функции (поле calls)
Time(s)		статистика времени выполнения функции в секундах
	Total	общее время, затраченное на эту функцию и все другие вызванные ей функции (поле total_time)
	Self	общее время, затраченное на выполнение самой функции, без учета других вызванных ей функций (поле self_time)
	Mean	среднее время выполнения одной функции
	Mean self	среднее собственное время выполнения одной функции

4.5.3. Top trigger functions by total time (Топ триггерных функций по общему времени)

Таблица 4.34 содержит топ триггерных функций, отсортированных по затраченному времени.

DB	Schema	Function	Executions	Time (s)			
				Total	Self	Mean	Mean self
jatoba	profile	grow_table_trg_f	150	15.16	15.16	0.101	0.101

Рисунок 4.34 – Пример отчета Top trigger functions by total time

Таблица 4.33 – Описание параметров отчета Top trigger functions by total time

Параметр		Описание
Database		имя БД функции
Schema		имя схемы индекса
Function		имя функции
Executions		количество вызовов данной функции (поле calls)
Time (s)		статистика времени выполнения функции в секундах
	Total	общее время, затраченное на эту функцию и все другие вызванные ей функции (поле total_time)
	Self	общее время, затраченное на выполнение самой функции, без учета других вызванных ей функций (поле self_time)
	Mean	среднее время выполнения одной функции
	Mean self	среднее собственное время выполнения одной функции

4.6. Vacuum-related stats (Статистика, связанная с вакуумом)

Данный раздел отчета содержит статистику работы процесса autovacuum. Если во время работы между снапшотами не было вакуумных операций – данный раздел останется пустым.

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

4.6.1. Top tables by vacuum operations (Топ таблиц по статистике вакуума)

Таблица 4.35 содержит топ таблицы, отсортированные по очистке (ручной и автоматической).

DB	Tablespace	Schema	Table	Vacuum count	Autovacuum count	Ins	Upd	Del	Upd(HOT)
jatoba	pg_default	pg_catalog	pg_attribute		2	140512		140500	
jatoba	pg_default	pg_catalog	pg_statistic		2	97500	315	97500	186

Рисунок 4.35 – Пример отчета Top tables by vacuum operations

Таблица 4.34 – Описание параметров отчета Top tables by vacuum operations

Параметр	Описание
Database	имя БД таблицы
Tablespace	имя табличного пространства, в котором находится таблица
Schema	имя схемы таблицы
Table	имя таблицы
Vacuum count	количество раз, когда эта таблица была обработана вручную (не считая VACUUM FULL) (поле vacuum_count)
Autovacuum count	количество раз, когда эта таблица была обработана фоновым процессом autovacuum (поле autovacuum_count)
Ins	количество вставленных строк (поле n_tup_ins)
Upd	количество обновленных строк (включая HOT) (поле n_tup_upd)
Del	количество удаленных строк (поле n_tup_del)
Upd(HOT)	количество обновленных строк HOT (поле n_tup_hot_upd)

4.6.2. Top tables by analyze operations (Топ таблиц по операциям анализа)

Таблица 4.36 содержит топ таблицы, отсортированные по количеству запусков анализа (ручных и автоматических).

DB	Tablespace	Schema	Table	Analyze count	Autoanalyze count	Ins	Upd	Del	Upd(HOT)
jatoba	pg_default	profile	last_stat_tables	1	1	160		162	
jatoba	pg_default	profile	last_stat_indexes	1	1	212		214	

Рисунок 4.36 – Пример отчета Top tables by analyze operations

Таблица 4.35 – Описание параметров отчета Top tables by analyze operations

Параметр	Описание
Database	имя БД таблицы
Tablespace	имя табличного пространства, в котором находится таблица
Schema	имя схемы таблицы
Table	имя таблицы
Analyze count	количество раз, когда эта таблица была обработана вручную (поле analyze_count)
Autoanalyze count	количество раз, когда эта таблица была обработана фоновым процессом autovacuum (поле autoanalyze_count)

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

Параметр	Описание
Ins	количество вставленных строк (поле n_tup_ins)
Upd	количество обновленных строк (включая HOT) (поле n_tup_upd)
Del	количество удаленных строк (поле n_tup_del)
Upd(HOT)	количество строк, обновленных по HOT (поле n_tup_hot_upd)

4.6.3. Top indexes by estimated vacuum load (Топ индексов по расчетной нагрузке вакуумного ввода/вывода)

В таблице 4.37 представлена оценка неявной вакуумной нагрузки, вызванной индексами таблицы. Здесь представлены топ индексы, отсортированные по количеству вакуумов, выполненные для базовой таблицы, умноженные на размер индекса.

DB	Tablespace	Schema	Table	Index	~Vacuum bytes	Vacuum cnt	Autovacuum cnt	IX size	Relsize
jatoba	pg_default	pg_catalog	pg_attribute	pg_attribute_relid_attnam_index	17 MB		2	8736 kB	768 kB
jatoba	pg_default	pg_catalog	pg_attribute	pg_attribute_relid_attnum_index	11 MB		2	5600 kB	768 kB

Рисунок 4.37 – Пример отчета Top indexes by estimated vacuum I/O load

Таблица 4.36 – Описание параметров отчета Top indexes by estimated vacuum load

Параметр	Описание
Database	имя БД индекса
Tablespace	имя табличного пространства, в котором расположен индекс
Schema	имя схемы индекса
Table	имя таблицы
Index	имя индекса
~Vacuum bytes	оценка вакуумной нагрузки, рассчитываемая как (vacuum_count + autovacuum_count) * index_size
Vacuum cnt	количество раз, когда эта таблица была обработана вручную (не считая VACUUM FULL) (поле vacuum_count)
Autovacuum cnt	количество раз, когда эта таблица была обработана фоновым процессом autovacuum (поле autovacuum_count)
IX size	средний размер индекса в течение интервала отчета
Relsize	средний размер отношения в течение интервала отчета

4.6.4. Top tables by dead tuples ratio (Топ таблиц по проценту наличия удаленных записей)

Данный раздел содержит измененные таблицы 4.38 с последним запуском vacuum. Статистика действительна для последнего снапшота в интервале отчета. Основана на представлении pg_stat_all_tables.

Топ таблицы размером 5 МБ и более, отсортированные по соотношению устаревших записей.

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

DB	Schema	Table	Live	Dead	%Dead	Last AV	Size
jatoba	pg_catalog	pg_statistic	1043	102	8	2021-09-17 02:00:35.549493-07	8768 kB

Рисунок 4.38 – Пример отчета Top tables by dead tuples ratio

Таблица 4.37 – Описание параметров отчета Top tables by dead tuples ratio

Параметр	Описание
Database	имя БД таблицы
Schema	имя схемы таблицы
Table	имя таблицы
Live	предполагаемое количество живых строк (n_live_tup)
Dead	предполагаемое количество мертвых строк (n_dead_tup)
%Dead	мертвые строки таблицы в процентах от всех строк таблицы
Last AV	последний раз, когда эта таблица была обработана фоновым процессом autovacuum (last_autovacuum)
Size	размер таблицы, каким он был на момент последнего снимка отчета

4.6.5. Top tables by modified tuples ratio (Топ таблиц по проценту измененных записей)

Данный раздел содержит модифицированные таблицы 4.39 с последним запуском vacuum. Статистика действительна для последнего снимка в интервале отчета. Основана на представлении pg_stat_all_tables.

Топ таблицы размером 5 МБ и более, отсортированные по соотношению модифицированных записей.

DB	Schema	Table	Live	Dead	Mod	%Mod	Last AA	Size
jatoba	pg_catalog	pg_statistic	1043	102	404545	35331		8768 kB

Рисунок 4.39 – Пример отчета Top tables by modified tuples ratio

Таблица 4.38 – Описание параметров отчета Top tables by modified tuples ratio

Параметр	Описание
Database	имя БД таблицы
Schema	имя схемы таблицы
Table	имя таблицы
Live	предполагаемое количество живых строк (n_live_tup)
Dead	предполагаемое количество мертвых строк (n_dead_tup)
Mod	предполагаемое количество строк, измененных с момента последнего анализа этой таблицы (n_mod_since_analyze)
%Mod	модифицированные строки таблицы в процентах от всех строк в таблице

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

Параметр	Описание
Last AA	последний раз, когда эта таблица была обработана фоновым процессом autovacuum
Size	размер таблицы, каким он был на момент последнего снапшота отчета

4.7. Cluster settings during the report interval (Настройка кластера во время отчетного интервала)

Данный раздел отчета содержит параметры СУБД «Jatoba», а также значения встроенных системных функций version(), pg_postmaster_start_time(), pg_conf_load_time() и поле system_identifier функции pg_control_system() в течение интервала отчета.

Defined settings				
Setting	reset_val	Unit	Source	Notes
TimeZone	America/Los_Angeles		/var/lib/jatoba/5/data/postgresql.conf:712	
config_file	/var/lib/jatoba/5/data/postgresql.conf			
data_directory	/var/lib/jatoba/5/data			
default_text_search_config	pg_catalog.english		/var/lib/jatoba/5/data/postgresql.conf:733	
hba_file	/var/lib/jatoba/5/data/pg_hba.conf			

Рисунок 4.40 – Пример отчета Cluster settings during the report interval

Таблица 4.39 – Описание параметров отчета Cluster settings during the report interval

Параметр	Описание
Setting	имя параметра
reset_val	поле reset_val представления pg_settings. Жирный шрифт используется для отображения настроек, измененных в течение интервала отчета
Unit	единица измерения параметра
Source	конфигурационный файл, в котором была определена данная настройка, номер строки после точки с запятой
Notes	поле, содержащее временную метку образца, когда это значение наблюдалось впервые в течение интервала отчета

4.8. Отчеты по компоненту «ja_Hipe_Cluster» (Citrus)

Данный раздел отчетов содержит параметры высокопроизводительного кластера, созданного компонентом «ja_Hipe_Cluster» (Citrus).

4.8.1. Nodes

Отчет содержит информацию об узлах кластера. В нем используются citus_stat_activity, pg_dist_node и функция run_command_on_all_nodes.

Таблица 4.40 - Описание параметров отчета Nodes

Название параметра	Название параметра в pg_profile	Тип данных параметра	Описание параметра
Nodeid	Node id	Integer	Id узла
Groupid	Group id	Integer	Id группы
Nodename	Node name	Text	Наименование узла
nodeport	Node port	Integer	Порт
noderole	Node role	Noderole	Является ли узел первичным или вторичным
nodecluster	Node cluster	Name	Имя кластера
available	Available	integer	Доступность узла
count_conn	Count connection	integer	Количество соединений

4.8.2. Connectivity between all nodes

Отчет содержит информацию о связь между всеми узлами кластера. В нем используется функция citus_check_cluster_node_health.

Таблица 4.41 – Описание параметров отчета Connectivity between all nodes

Название параметра	Название параметра в pg_profile	Тип данных параметра	Описание параметра
From_nodename	From node name	Text	Имя исходного узла
From_nodeport	From node port	integer	Порт на исходном узле
To_nodename	To node name	text	Имя узла назначения
To_nodeport	To node port	Integer	Порт на узле назначения
Available	Available	integer	Доступность узла

4.8.3. Tables (citus_tables)

Отчет содержит информацию о таблицах, распределенных и справочных.

Таблица 4.42 – Описание параметров отчета Tables

Название параметра	Название параметра в pg_profile	Тип данных параметра	Описание параметра
table_name	Table	Text	Наименование таблицы
citus_table_type	Type	Text	Тип таблицы (distributed - распределенная, reference - справочная)
distribution_column	Distribution column	Text	Наименование столбца распределения
colocation_id	Colocation id	Integer	Идентификатор группы совместного размещения
table_size	Table size	Text	Размер таблицы
shard_count	Shard count	Bigint	Число шардов
access_method	Access method	Text	Метод доступа (heap, columnar)
	Growth size	Text	Размер таблицы (разница между текущим и предыдущим снимками)
	Growth shard count	Bigint	Число шардов (разница между текущим и предыдущим снимками)

4.8.4. Shards

В отчет входит информация о том, к какой распределенной таблице принадлежит шард, где находится каждый шард (нода/узел и порт), к какому типу таблицы он принадлежит, его размер и статистика о столбце распределения для этого шарда. В случае хэш-распределенных таблиц это диапазоны хэш-токенов, назначенных этому шарду.

Представление citus_shards помогает изучать сегменты, позволяя, среди прочего, находить дисбаланс по размеру между узлами.

Таблица 4.43 – Описание параметров отчета Shards

Название параметра	Название параметра в pg_profile	Тип данных параметра	Описание параметра
citus_shards.table_name	Table	Regclass	Наименование таблицы и схемы
citus_shards.shardid	Shard id	Bigint	Id шарда
citus_shards.shard_name	Shard name	Text	Наименование шарда
citus_shards.citus_table_type	Table type	Text	Тип таблицы
citus_shards.colocation_id	Colocation id	Integer	Идентификатор группы совместного размещения
citus_shards.nodename	Node name	Text	Наименование ноды на которой расположен шард

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

Название параметра	Название параметра в pg_profile	Тип данных параметра	Описание параметра
citus_shards.nodeport	Node port	Integer	Порт
citus_shards.shard_size	Shard size	Bigint	Размер шарда
pg_dist_shard.shardstorage	Shard storage	character	Тип хранилища, используемый для этого шарда ('t', 'c', 'f').
pg_dist_shard.shardminvalue	Min value	Text	Для распределенных хэш-таблиц минимальное значение
pg_dist_shard.shardmaxvalue	Max value	text	Для распределенных хэш-таблиц максимальное значение
	Growth size	Bigint	Размер шарда (разница между текущим и предыдущим снимками)

4.8.5. Blocked queries (citus_lock_waits)

Отчет содержит информацию о запросах, заблокированных во всем кластере.

Таблица 4.44 – Описание параметров отчета Blocked queries

Название параметра	Название параметра в pg_profile	Тип данных параметра	Описание параметра
waiting_gpid	Waiting id	bigint	Id ожидающего запроса
blocking_gpid	Blocking id	bigint	Id блокирующего запроса
blocked_statement	Waiting query	Text	Заблокированный запрос
current_statement_in_blocking_process	Blocking query	Text	Блокирующий запрос
waiting_nodeid,	Waiting node id	Integer	Id ноды ожидающего запроса
blocking_nodeid	Blocking node id	integer	Id ноды блокирующего запроса

4.8.6. Query statistics (citus_stat_statements)

Отчет содержит статистику, как и для кого выполняются запросы.

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

Таблица 4.45 – Описание параметров отчета Query statistics

Название параметра	Название параметра в pg_profile	Тип данных параметра	Описание параметра
queryid	Query id	bigint	Id запроса
rolname	User	oid	Наименование пользователя выполнившего запрос
datname	Database	oid	Наименование бд
query	Query	text	Строка запроса
partition_key	Partition key	text	Значения столбца распределения в запросах
calls	Calls	bigint	Количество запусков запросов

4.8.7. Rebalance progress (get_rebalance_progress)

Отчет содержит информацию о процессе ребалансировки.

Таблица 4.46 - Описание параметров отчета Rebalance progress

Название параметра	Название параметра в pg_profile	Тип данных параметра	Описание параметра
Table_name	Table	Regclass	Наименование таблицы
Shardid	Shard id	Bigint	Id шарда
Source_shard_size	Source shard size	Text	Размер исходного шарда
Target_shard_size	Target shard size	Text	Размер целевого шарда
Percent_completed_estimate	Percent	numeric	Процент выполнения

4.8.8. Configuration parameters

Отчет о параметрах конфигурации.

Таблица 4.47 - Описание параметров отчета Configuration parameters

Название параметра	Название параметра в pg_profile	Тип данных параметра	Описание параметра
Name	name	Text	Наименование параметра конфигурации
Setting	setting	Text	Текущее значение параметра конфигурации

4.8.9. Active tenants (citus_stat_tenants)

В случае если установленная версия Citus меньше чем 11.3, то отчет по таблице Active tenants не отображается.

Таблица 4.48 – Описание параметров отчета Active tenants

Название параметра	Название параметра в pg_profile	Тип данных параметра	Описание параметра
Nodename	Node name	Text	Наименование ноды
Tenant_attribute	Tenant attribute	Text	Значение в столбце распределения, идентифицирующее tenant
Read_count_in_this_period	Read count in this period	Int	Количество запросов на чтение (select) за период
Read_count_in_last_peroid	Read count in last peroid	Int	Количество запросов чтения за прошлый период
Query_count_in_this_period	Query count in this period	Int	Количество запросов на чтение/запись для tenant за период времени
Query_count_in_last_period	Query count in last period	int	Количество запросов на чтение/запись для tenant за прошлый период
Cpu_usage_in_this_period	Cpu usage in this period	Float	Секунды процессорного времени, затраченные этим tenant за период
Cpu_usage_in_last_period	Cpu usage in last period	float	Секунды процессорного времени, затраченные этим tenant за прошлый период

5. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ


1) СУБД собирает статистику после завершения выполнения запроса. Если выполнение запроса длится в течение нескольких снапшотов, это повлияет на статистику только последнего снапшота (когда он был снят). И получить статистику по все еще выполняющимся операторам будет нельзя. Процессы обслуживания, такие как vacuum и checkpoint, обновляют статистику только после завершения.

2) Сброс любой статистики СУБД может повлиять на корректность формирования последующих запросов (планировщик запросов опирается на данные статистики при выборе методов обхода таблиц, использования индексов и т.д.).

3) Эксклюзивные блокировки на таблицах конфликтуют с процедурой вычисления размера таблицы. Снапшот не будет собирать размеры таблиц с AccessExclusiveLock, принадлежащим какой-либо сессии. Сессия может получить AccessExclusiveLock на таблицу во время обработки выборки. Для того, чтобы обойти эту проблему, lock_timeout (настройка СУБД «Jatoba», отвечающая за максимальное разрешенное время блокировки таблицы/индекса, строки или иного объекта БД) установлен на 3 секунды, поэтому если функция take_sample() не сможет получить блокировку в течение 3 секунд, она завершится неудачей, и снапшот не будет снят.

ТЕРМИНЫ И ОПРЕДЕЛЕНИЕ

AccessExclusiveLock	вид блокировки таблицы, гарантирующий, что доступ на момент блокировки имеет только та транзакция, которая вызвала блокировку
baseline	именованная последовательность снапшотов, которая имеет отдельную от настроенной политику хранения. Можно задать как определенное время хранения в днях, так и бесконечное время хранения, оставив соответственный параметр пустым. Также можно создать последовательность снапшотов только для определенного периода времени, например, в пиковые часы загрузки
dblink	расширение, позволяющее выполнить запрос к удаленной БД. Распространяется в составе СУБД PostgreSQL/Jatoba
HOT (Heap-only Tuple) обновление	механизм оптимизации выполнения команд UPDATE по изменению строк таблиц в СУБД. Данный механизм добавляет новую версию строки рядом со старой версией строки на той же странице данных и устанавливает соответствующие флаги следования записей в порядке их обновления. При таком подходе не нужно обновлять все индексы таблицы и записывать в них местоположение новой строки на диске. Данный механизм работает только для полей таблиц, не входящих в индексы, и призван уменьшить нагрузку на файловую систему
lock_timeout	параметр, задающий максимально возможное время блокировки таблицы / индекса / строки

<code>pg_conf_load_time()</code>	функция СУБД, которая возвращает данные о времени (моменте) загрузки конфигурации СУБД
<code>pg_control_system()</code>	функция СУБД, которая возвращает информацию о текущем состоянии управляющего файла СУБД
<code>pg_stat_kcache</code>	расширение, собирающее статистику по операциям чтения и записи на уровне файловой системы. Распространяется отдельно от СУБД PostgreSQL/Jatoba
	 Работает только на Linux системах
<code>pg_stat_kcache.track_planning = on/off</code>	параметр расширения <code>pg_stat_kcache</code> , контролирующий сбор данных об операциях планирования SQL-запросов и времени планирования
<code>pg_postmaster_start_time()</code>	функция СУБД, которая возвращает данные о времени (моменте) запуска СУБД
<code>pg_stat_statements</code>	расширение, собирающее статистику выполнения SQL-запросов на сервере БД. Распространяется в составе СУБД PostgreSQL/Jatoba
<code>plpgsql</code>	расширение языка SQL (процедурный язык), используемое в СУБД PostgreSQL/Jatoba. Этот язык предназначен для написания хранимых процедур и функций
<code>postgresql.conf</code>	файл конфигурации СУБД
Statistic Collector (сборщик статистики)	встроенный в СУБД механизм, позволяющий собирать метрики активности сервера БД
<code>vacuum</code> (вакуум)	механизм СУБД, удаляющий старые версии строк таблиц в БД

version() и jatoba_version()

функции СУБД, которые возвращают информацию о версии СУБД

снапшот

снимок состояния БД на конкретный момент времени

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ID	–	Identifier
SQL	–	Structured Query Language
БД	–	База данных
ОС	–	Операционная система
СУБД	–	Система управления базами данных

[illegible]

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