

АО «Концерн ГРАНИТ»

АКЦИОНЕРНОЕ ОБЩЕСТВО



УТВЕРЖДАЮ

Генеральный директор

АО «Концерн ГРАНИТ»



С.С. Гостев

« 18 » июня 2024 г.

СУБД «Квант-Гибрид 1.5»

Руководство пользователя

ВЕР.00207-01 33 01

Листов 31

АННОТАЦИЯ

Настоящий документ содержит информацию по основным полномочиям пользователей и принципам работы с объектно-реляционной системой управления базами данных «Квант-Гибрид 1.5» (далее – СУБД «Квант-Гибрид 1.5»).

СОДЕРЖАНИЕ

1	ОБЩИЕ СВЕДЕНИЯ	4
1.1	Назначение	4
1.2	Функции.....	4
1.3	Условия выполнения	4
1.4	Режимы функционирования	6
2	СТРУКТУРА	7
2.1	Сведения о составных частях Системы (ПО).....	7
2.2	Сведения о связях с другими программами	7
3	ПОЛНОМОЧИЯ ПОЛЬЗОВАТЕЛЕЙ.....	8
3.1	Роли мониторинга сервера базы данных	9
3.2	Доверенная роль pg_signal_backend	9
3.3	Расширенная пользовательская роль.....	9
3.4	Пользовательская роль.....	10
4	РАБОТА С СУБД	11
4.1	Порядок работы с СУБД.....	12
4.1.1	Подключение к СУБД.....	13
4.1.2	Сообщения об ошибках при аутентификации.....	14
4.1.3	Сообщения об ошибках при подключении клиентских приложений к СУБД	14
4.2	Работа с данными	15
4.2.1	Запись данных в базу	15
4.2.2	Чтение данных	16
4.2.3	Редактирование существующих записей.....	17
4.2.4	Удаление записей	18
4.3	Принципы безопасной работы СУБД.....	19
4.4	Очистка данных в СУБД.....	20
5	СООБЩЕНИЯ ПОЛЬЗОВАТЕЛЯМ	26
5.1	Поля сообщений с ошибками и замечаниями	26
	СОКРАЩЕНИЯ И ОПРЕДЕЛЕНИЯ.....	29

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение

СУБД «Квант-гибрид 1.5» представляет собой цифровую платформу, применимую для широкого круга разработчиков государственных информационных систем, приложений, а также для корпоративных отделов по цифровой трансформации предприятий крупного и среднего бизнеса.

1.2 Функции

СУБД «Квант-гибрид 1.5» обеспечивает выполнение следующих функций:

- 1) Организация необходимой структуры базы данных (Инициализация экземпляра базы данных и запуск сервера баз данных).
- 2) Создание и удаление экземпляра базы данных.
- 3) Создание и удаление новой учетной записи.
- 4) Запись данных в базу, обеспечение записи данных, вводимых пользователем в базу данных через интерактивный терминал `psql`.
- 5) Управление хранением данных.
- 6) Чтение данных (выполнение запросов пользователя на получение интересующих данных).
- 7) Редактирование существующих записей.
- 8) Очистка устаревших копий записей.
- 9) Удаление записей.
- 10) Реализация поддержки языка описания данных и языка запросов.
- 11) Очистка базы данных и генерация внутренней статистики.
- 12) Обеспечение совместного использования объектов базы данных.
- 13) Обеспечение возможности прямой загрузки данных.
- 14) Организация сбора, агрегации и пересылки метрик в различные системы.
- 15) Контроль доступа к объектам БД:
 - a. Ролевой метод управления доступом.
 - b. Дискреционный метод управления доступом.

- 16) Идентификация и аутентификация.
- 17) Поддержка целостности и доступности данных:
 - a. Резервное копирование и восстановление кластеров баз данных.
 - b. Управление обновлениями.
 - c. Предоставление возможности использовать асинхронный пул соединений.
 - d. Обеспечение целостности данных.
 - e. Подсчет, проверка контрольных сумм хранимых данных СУБД.
 - f. Криптографическая защита данных.
- 18) Регистрация событий.
- 19) Очистка памяти.

1.3 Условия выполнения

СУБД «Квант-гибрид 1.5» функционирует на ПЭВМ с характеристиками, не ниже следующих:

- процессор архитектур: x86-64 с тактовой частотой 1,8 ГГц;
- оперативная память: не менее 4 Гбайт оперативной памяти;
- жёсткий диск: не менее 400 МБ (не учитывая размер базы данных);
- устройство чтения компакт-дисков;
- звуковая карта;
- сетевая плата Fast Ethernet 100 Мбит/с.

СУБД «Квант-гибрид 1.5» функционирует в ОС на базе Linux:

- операционная система Альт 8 СП (обозначение программного изделия — ЛКНВ.11100-01) (№ сертификата 3866);
- операционная система специального назначения «Astra Linux Special Edition» (в исполнении РУСБ.10015-01) x86_64, очередное обновление 1.7, (№ сертификата 2557);
- операционная система «РЕД ОС» (№ сертификата 4060).

1.4 Режимы функционирования

Функционирование СУБД «Квант-гибрид 1.5» предусматривает следующие режимы работы:

- Нормальный режим функционирования компонента, включающий в себя:
 - обеспечение полной готовности СУБД «Квант-гибрид 1.5»;
 - обеспечение круглосуточного (24*7) функционирования СУБД «Квант-гибрид 1.5»;
 - обеспечение исправной работы оборудования, составляющего комплекс технических средств (включая каналы связи).
- Аварийный режим функционирования. СУБД «Квант-гибрид 1.5» поддерживает аварийный режим функционирования – режим аварийного отказа. В этом случае резервные средства контроля и управления должны обеспечивать безопасную остановку СУБД «Квант-гибрид 1.5» для последующего восстановления работоспособности системы силами дежурного персонала в установленные показателями назначения сроки. При восстановлении работы СУБД «Квант-гибрид 1.5» продолжает работу с места останова или с места начала обработки данных, которая не была завершена.
- Технологический режим функционирования. СУБД «Квант-гибрид 1.5» поддерживает технологический режим функционирования, обеспечивающий конфигурирование, перенастройку или техническое обслуживание СУБД «Квант-гибрид 1.5» и её баз данных с возможным частичным снижением быстродействия и блокировкой отдельных функциональных возможностей.

2 СТРУКТУРА

СУБД «Квант-гибрид 1.5» реализована в архитектуре клиент-сервер и представляет собой общесистемное ПО, обслуживающее запросы клиентов на языке SQL. Приложения пользователя (клиенты создаются сторонними разработчиками), желающее исполнить операции на языке SQL должны взаимодействовать с сервером Квант-гибрид по сетевому протоколу, или при помощи механизмов межпроцессного взаимодействия операционной системы сервера Квант-гибрид.

2.1 Сведения о составных частях Системы (ПО)

ПО неделимо, все взаимодействия происходят на уровне ядра Системы. С внешними узлами и компонентами ПО не взаимодействует.

2.2 Сведения о связях с другими программами

ПО взаимодействует с клиентами по протоколу TCP/IP.

3 ПОЛНОМОЧИЯ ПОЛЬЗОВАТЕЛЕЙ

В СУБД «Квант-гибрид 1.5» предоставлен по умолчанию набор ролей, которые обеспечивают доступ к специальным возможностям и информации БД. Администраторы могут предоставлять эти роли пользователям и/или другим ролям, тем самым расширяя доступы к указанным возможностям и информации.

Таблица 1 – Роли по умолчанию

Роль	Разрешенный доступ
pg_read_all_settings	Чтение всех переменных конфигураций, даже тех, которые обычно видны только суперпользователям.
pg_read_all_stats	Чтение всех представлений pg_stat_* и использование различных статистических расширений, даже тех, которые обычно видны только суперпользователям.
pg_stat_scan_tables	Выполнение функций мониторинга, которые могут блокировать ACCESS SHARE для таблиц, возможно, в течение длительного времени.
pg_monitor	Чтение/выполнение различных видов мониторинга и функций. Эта роль является членом групп pg_read_all_settings, pg_read_all_stats и pg_stat_scan_tables.
pg_database_owner	Никакого. Членство неявно включает в себя текущего владельца базы данных.
pg_signal_backend	Подавать другому обслуживающему процессу сигнал, чтобы отменить запрос или завершить сеанс.
pg_read_server_files	Чтение файлов из любого места, к которому база данных может получить доступ на сервере с помощью COPY и других функций доступа к файлам.
pg_write_server_files	Запись в файлы в любом месте, к которому база данных может получить доступ на сервере с помощью COPY и других функций доступа к файлам.
pg_execute_server_program	Выполнение программ на сервере базы данных как пользователь, база данных запускается так же, как с COPY и другими функциями, которые позволяют выполнять программу на стороне сервера.
pg_checkpoint	Разрешить выполнение команды CHECKPOINT.

У роли `pg_database_owner` имеется только один неявный, определяемый сложившейся ситуацией член, а именно — владелец текущей базы данных. Изначально эта роль не предоставляет никаких прав. Как и любая роль, она может владеть объектами или получать права доступа. Следовательно, как только `pg_database_owner` приобретет права в базе-шаблоне, каждый владелец базы данных, созданной из этого шаблона, получит эти права. Роль `pg_database_owner` не может быть членом какой-либо роли и не может явно включать членов. Эта роль владеет схемой `public`, так что каждый владелец базы данных управляет локальным использованием этой схемы.

3.1 Роли мониторинга сервера базы данных

Роли `pg_monitor`, `pg_read_all_settings`, `pg_read_all_stats` и `pg_stat_scan_tables` предназначены для того, чтобы администраторы могли легко настроить роль для мониторинга сервера базы данных. Они предоставляют набор общих полномочий, позволяющих читать различные полезные параметры конфигурации, статистику и другую системную информацию, обычно доступную только суперпользователям.

3.2 Доверенная роль `pg_signal_backend`

Роль `pg_signal_backend` предназначена для того, чтобы администраторы могли включать доверенные роли (не роль суперпользователя) для отправки сигналов другим процессам. В настоящее время эта роль позволяет отправлять сигналы для отмены запроса на другом сервере или завершении его сеанса. Однако пользователь, которому предоставлена эта роль, не может отправлять сигналы процессу, принадлежащему суперпользователю.

3.3 Расширенная пользовательская роль

Роли `pg_read_server_files`, `pg_write_server_files` и `pg_execute_server_program` предназначены для того, чтобы администраторы могли иметь доверенные роли (не роль суперпользователя), которые могут получать доступ к файлам и запускать программы на сервере базы данных в качестве пользователя базы данных. Эти роли могут получить доступ к любому файлу в файловой системе сервера, они пропускают

все проверки разрешений на уровне базы данных при непосредственном доступе к файлам и могут использоваться для получения доступа на уровне суперпользователя.

3.4 Пользовательская роль

Пользовательские роли могут иметь различные наименования и позволяют пользователям работать с базой данных (читать и записывать данные в таблицы баз данных), без предоставления им расширенных полномочий п 3.1-3.3. Типовые варианты ролей пользователя СУБД:

- аналитик БД – чтение данных из СУБД;
- разработчик приложения – модификация структуры БД, разработка функций и настройка БД для подключения внешнего приложения;
- пользовательское приложение – чтение и запись данных в настроенной структуре БД.

Параметры роли могут настраиваться администратором. Более подробное описание ролевой системы представлено в документе ВЕМР.00207-01 32 01 «Руководство системного администратора».

Кроме данных ролей в Системе существуют, роли администраторов информационной безопасности: *Администратор СУБД* и *Администратор БД*. Наличие в системе ролей *Администраторов СУБД* и *Администраторов конкретных БД* позволяет назначать Администраторов Информационной Безопасности не включая атрибут Суперпользователя у соответствующих записей в таблице каталога пользователей и ролей *pg_authid*.

4 РАБОТА С СУБД

Для работы с СУБД «Квант-гибрид 1.5» необходимо предварительно ее установить и выполнить запуск СУБД силами администратора СУБД. Для получения доступа к СУБД требуется обратиться к администратору СУБД.

Клиенты СУБД (приложения и пользователи) могут обращаться к кластеру СУБД через интерфейс `libpq`. `Libpq` – это набор библиотечных функций, позволяющих передавать запросы внутреннему серверу СУБД и получать результаты этих запросов. В состав СУБД входит утилита ***psql***, работающая через `libpq`. Утилита ***psql*** представляет собой терминальный клиент для работы с СУБД (командный интерфейс для работы с СУБД).

Клиенты СУБД могут обращаться к кластеру СУБД посредством элементов, реализованных в виде модуля с набором утилит и обозначенных на схеме как блок «Команды БД и утилиты» (в данный блок входят такие элементы, как `dropdb`, `createdb`, `initdb`, `psql` и другие). Элементы из блока «Команды БД и утилиты» могут взаимодействовать непосредственно с кластером СУБД или взаимодействовать через интерфейс `libpq`.

Существует возможность работы через «Асинхронный пул соединений» (Quantum Connection Pool, QCP), который представляет собой балансировщик сетевой нагрузки, предназначенный для оптимального использования серверных подключений. Этот модуль обеспечивает подключение и поддержку связи с удаленными клиентами.

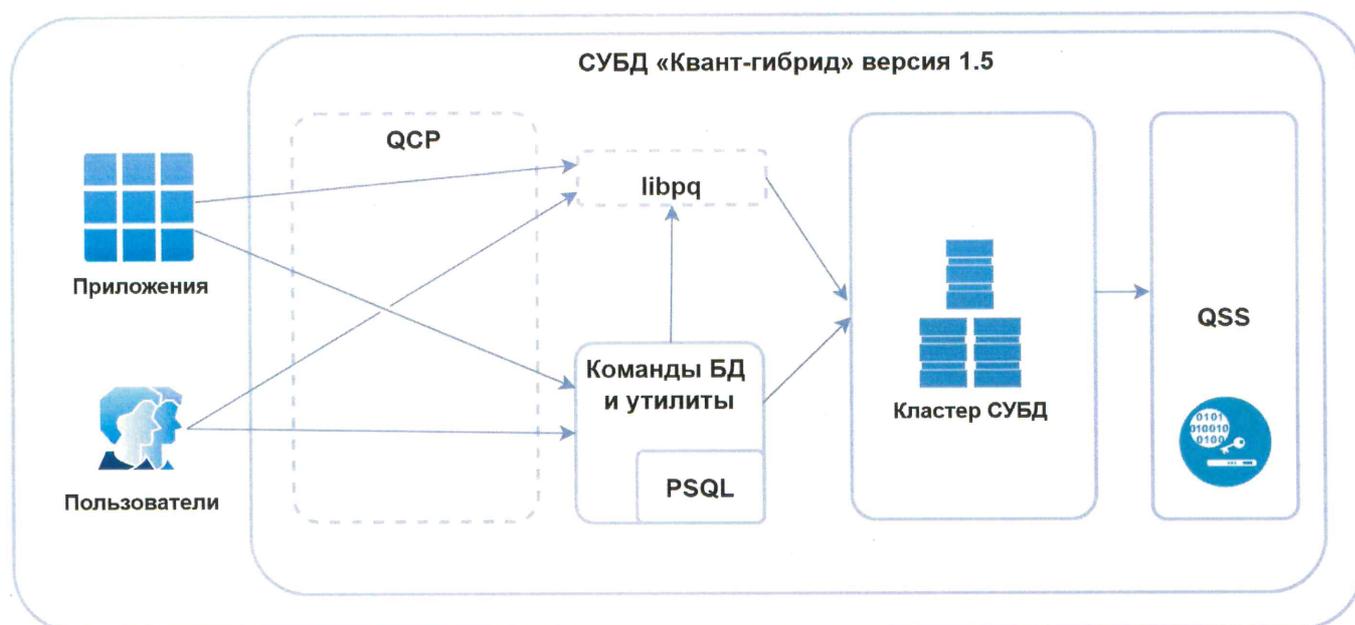


Рисунок 1 - Структурная схема Системы

4.1 Порядок работы с СУБД

В зависимости от настроек СУБД и сервера, на котором она расположена возможна работа на сервере или удаленное подключение.

Для работы с СУБД, пользователь должен:

- 1) Авторизоваться на устройстве, с которого будет производиться подключение, используя учетную запись пользователя ОС (роль пользователя, роль администратора). Например, на сервер, на котором расположена СУБД.
- 2) Выполнить проверку наличия на сервере запущенной СУБД. Для этого необходимо в строке терминала ввести команду: `systemctl status qhb`.
- 3) Выполнить подключение к нужной БД, используя учетную запись пользователя СУБД. Для этого необходимо в командной строке терминала ввести: `psql [параметр...] [имя_бд [имя_пользователя]]`.

Более подробно данный пункт представлен в разделе «Подключение к СУБД» настоящего документа.

- 4) После этого, пользователь будет переведен в оболочку `psql` для взаимодействия с БД. Затем пользователю будут доступны операции с СУБД в зависимости от его роли и полномочий. Расширенное описание

доступных команд и параметров для `psql` приведено в технической документации по СУБД «Квант-Гибрид» 1.5.1 на ресурсе разработчика.

5) Для выхода из `psql`, в строке терминала ввести команду: `"\q"`.

4.1.1 Подключение к СУБД

Для подключения к базе данных необходимо знать имя целевой базы, имя хоста и номер порта сервера, а также имя пользователя БД. Для `psql` эти параметры можно задать через аргументы командной строки, а именно `-d`, `-h`, `-p` и `-U` соответственно. При обнаружении в командной строке аргумента, не относящегося ни к одному параметру, оно будет воспринято как имя базы данных (или имя пользователя, если имя базы данных уже задано). Задавать все эти аргументы необязательно: у них есть значения по умолчанию. Если опустить имя хоста, `psql` подключится через Unix-сокеты к серверу на локальном хосте или через TCP/IP к `localhost`, если в системе нет Unix-сокетов. Номер порта по умолчанию определяется во время компиляции. Поскольку сервер баз данных использует то же значение по умолчанию, в большинстве случаев указывать порт нет необходимости. Имя пользователя по умолчанию, как и имя базы данных по умолчанию — это имя пользователя в операционной системе. Подключиться к любой базе данных под любым именем пользователя не получится. О правах доступа, которые назначены конкретному пользователю, можно узнать у администратора баз данных.

Если значения по умолчанию не подходят, вместо ввода нужных параметров вручную можно установить переменные среды `PGDATABASE`, `PGHOST`, `PGPORT` и/или `PGUSER` в подходящие значения.

Альтернативный способ указания параметров подключения — использование вместо имени базы данных строки `conninfo` или `URI`. Этот способ также позволяет использовать для получения параметров подключения `LDAP`.

Если по какой-либо причине соединение не может быть установлено (например, недостаточно прав, сервер на целевом хосте не работает и т.д.), `psql` вернет ошибку и прекратит работу.

Если и стандартный ввод, и стандартный вывод являются терминалом, то `psql` установит кодировку клиента в `«auto»`, и подходящая клиентская кодировка будет

определяться из настроек параметров пользовательского интерфейса (переменная среды LC_STYPE в системах Unix). Если это работает не так, как ожидалось, клиентскую кодировку можно изменить, установив переменную среды PGCLIENTENCODING.

4.1.2 Сообщения об ошибках при аутентификации

Сбои и другие проблемы с аутентификацией обычно проявляют себя посредством сообщений об ошибках, подобных следующему:

```
FATAL: no qhb_hba.conf entry for host "123.123.123.123", user "andym", database "testdb"
```

Это сообщение выводится, если связь с сервером есть, но он отказал в подключении. В сообщении содержится предположение, что сервер отказал в подключении, потому что не может найти соответствующую запись в своем файле конфигурации qhb_hba.conf.

```
FATAL: password authentication failed for user "andym"
```

Подобные сообщения означают, что вы связались с сервером, но продолжение взаимодействия с сервером возможно только после того, как будет пройдена авторизацию по методу, указанному в файле qhb_hba.conf. Проверьте вводимый пароль либо настройки Kerberos или Ident, если в сообщении упоминается один из этих типов аутентификации.

```
FATAL: user "andym" does not exist
```

Указанное имя пользователя базы данных не найдено.

```
FATAL: database "testdb" does not exist
```

Сообщение означает, что базы данных, к которой вы пытаетесь подключиться, не существует. Если имя базы данных не было указано при подключении, то по умолчанию берется имя пользователя базы данных, что может привести к ошибке.

4.1.3 Сообщения об ошибках при подключении клиентских приложений к СУБД

Ошибки на стороне клиента зависят от приложения, но некоторые из них могут быть связаны с тем, как был запущен сервер. Условия, отличные от указанных ниже, должны быть задокументированы в соответствующем клиентском приложении.

```
psql: could not connect to server: Connection refused
```

Is the server running on host "server.joe.com" and accepting TCP/IP connections on port 5432?

Общая ошибка «Не могу найти сервер для связи» может появиться при попытке установить связь TCP/IP. Распространенная причина ошибки: не включены настройки сервера для разрешения соединений по TCP/IP. В этом случае необходимо обратиться к администратору.

В качестве альтернативы, при попытке соединения с локальным сервером с использованием доменных сокетов Unix может быть выведено сообщение:

psql: could not connect to server: No such file or directory

Is the server running locally and accepting connections on Unix domain socket "/tmp/.s.PGSQL.5432"?

Последняя строка поможет проверить факт того, что клиент пытается подключиться к заданному серверу. Если не работает ни один сервер, сообщение об ошибке будет либо «*Connection refused*» либо «*No such file or directory*», в этом контексте «*Connection refused*» в соединении, не означает, что сервер получил ваш запрос на соединение и отклонил его. В этом случае появится другое сообщение, которое указывает на более фундаментальные проблемы: такие как отсутствие сетевого подключения.

4.2 Работа с данными

4.2.1 Запись данных в базу

Для записи данных в базу данных и обеспечения корректности записи данных, вводимых пользователем через интерактивный терминал psql, пользователь должен выполнить следующие действия:

- 1) Войти в операционную систему.
- 2) Запустить интерактивный терминал psql, используя команду psql с необходимыми параметрами, включая параметр -h для удаленного подключения.
- 3) Выбрать целевую базу данных с помощью команды "\c имя_базы_данных", чтобы переключиться на нужную базу данных для выполнения операций с данными.

- 4) Выполнить SQL-запросы для вставки, обновления или удаления данных в соответствующих таблицах базы данных. Запросы должны быть сформулированы с учетом правил и ограничений, заданных администратором СУБД.

Например:

- Вставка данных: *INSERT INTO table_name (column1, column2, column3) VALUES (value1, value2, value3);*
- Обновление данных: *UPDATE table_name SET column1 = value1, column2 = value2 WHERE condition;*
- Удаление данных: *DELETE FROM table_name WHERE condition;*

Проверить результаты выполненных запросов, используя команды SELECT для получения данных из таблиц. Например:

*SELECT * FROM table_name WHERE condition;*

4.2.2 Чтение данных

Для чтения данных из базы данных и выполнения запросов на получение интересующей информации, пользователь должен выполнить следующие действия:

- 1) Войти в операционную систему, используя учетные данные администратора или пользователя с соответствующими правами доступа.
- 2) Запустить интерактивный терминал psql, используя команду psql с необходимыми параметрами, включая параметр -h для удаленного подключения и параметры аутентификации, такие как имя пользователя (-U) и пароль (-W).
- 3) Выбрать целевую базу данных с помощью команды "\c имя_базы_данных", чтобы переключиться на нужную базу данных для выполнения операций с данными.
- 4) Выполнить SQL-запросы для извлечения данных из соответствующих таблиц базы данных. Запросы должны быть сформулированы с учетом

правил и ограничений, заданных администратором СУБД. Примеры запросов на чтение данных:

- Выбрать все данные из таблицы: *SELECT * FROM table_name;*
- Выбрать определенные столбцы из таблицы: *SELECT column1, column2 FROM table_name;*
- Выбрать данные с условием: *SELECT * FROM table_name WHERE condition;*
- Сортировка данных: *SELECT * FROM table_name ORDER BY column1 ASC, column2 DESC;*
- Группировка данных и агрегатные функции: *SELECT column1, COUNT(*) FROM table_name GROUP BY column1 HAVING COUNT(*) > 1;*

- 5) Просмотреть результаты выполненных запросов в интерактивном терминале `psql`. Результаты будут отображаться в табличном виде, и пользователь может анализировать полученную информацию.
- 6) При необходимости, сохранить результаты запросов в файл или экспортировать данные в другие форматы с использованием соответствующих команд и утилит.
- 7) Завершить работу с интерактивным терминалом `psql`, используя команду `\q`.

4.2.3 Редактирование существующих записей

Для редактирования существующих записей в базе данных, пользователь должен выполнить следующие действия:

- 1) Войти в операционную систему, используя учетные данные администратора или пользователя с соответствующими правами доступа.
- 2) Запустить интерактивный терминал `psql`, используя команду `psql` с необходимыми параметрами, включая параметр `-h` для удаленного подключения и параметры аутентификации, такие как имя пользователя (`-U`) и пароль (`-W`).

- 3) Выбрать целевую базу данных с помощью команды "\c имя_базы_данных", чтобы переключиться на нужную базу данных для выполнения операций с данными.
- 4) Выполнить SQL-запросы для обновления данных в соответствующих таблицах базы данных. Запросы должны быть сформулированы с учетом правил и ограничений, заданных администратором СУБД. Примеры запросов на редактирование существующих записей:
 - Обновление одного поля:
UPDATE имя_таблицы SET поле1 = новое_значение WHERE условие;
 - Обновление нескольких полей:
UPDATE имя_таблицы SET поле1 = новое_значение1, поле2 = новое_значение2 WHERE условие;
- 5) После выполнения SQL-запросов на редактирование записей, убедитесь, что изменения были успешно применены, сделав запрос на чтение данных и проверив результаты. Например, используйте команду SELECT для проверки обновленных данных:
*SELECT * FROM имя_таблицы WHERE условие;*

4.2.4 Удаление записей

Для удаления записей из базы данных, пользователь должен выполнить следующие действия:

- 1) Войти в операционную систему, используя учетные данные администратора или пользователя с соответствующими правами доступа.
- 2) Запустить интерактивный терминал `psql`, используя команду `psql` с необходимыми параметрами, включая параметр `-h` для удаленного подключения и параметры аутентификации, такие как имя пользователя (`-U`) и пароль (`-W`).
- 3) Выбрать целевую базу данных с помощью команды "\c имя_базы_данных", чтобы переключиться на нужную базу данных для выполнения операций с данными.

4) Выполнить SQL-запросы для удаления данных из соответствующих таблиц базы данных. Запросы должны быть сформулированы с учетом правил и ограничений, заданных администратором СУБД. Примеры запросов на удаление записей:

– Удаление записей с определенным условием:

DELETE FROM имя_таблицы WHERE условие;

– Удаление всех записей из таблицы:

DELETE FROM имя_таблицы;

5) После выполнения SQL-запросов на удаление записей, убедитесь, что изменения были успешно применены, сделав запрос на чтение данных и проверив результаты. Например, используйте команду SELECT для проверки удаленных данных:

*SELECT * FROM имя_таблицы WHERE условие;*

4.3 Принципы безопасной работы СУБД

Необходимо разделять пользовательские учетные записи СУБД и учетные записи операционной системы, на которой СУБД установлена. Администраторы СУБД и пользователи СУБД должны работать в операционной системе под учетными записями с ограниченными правами (пользователи могут не иметь учетных записей в операционной системе: рекомендуется подключаться к серверу с СУБД удаленно), для обеспечения невозможности влияния на загрузку СУБД и работу функций безопасности СУБД, в том числе оказания влияния на данные журналов регистрации событий. Администратор СУБД и пользователь операционной системы с правами администратора для безопасной работы СУБД должны быть разными лицами.

При указанной в предыдущем параграфе настройке операционной системы, настройки и параметры безопасности пользователям СУБД (за исключением администратора) недоступны, а у администратора СУБД ограничены. Настройки и параметры безопасности, доступные администратору, указаны в документе «Руководство системного администратора».

Несмотря на то, что интерфейсы для работы с СУБД едины для всех пользователей СУБД (включая суперпользователей), пользовательские полномочия различны и обусловлены разграничением прав доступа пользователей к объектам баз данных на основе правил разграничения доступа. Подробнее об этом написано в разделе 3 настоящего руководства и в документе «Руководство системного администратора».

При обнаружении сбоев СУБД пользователям необходимо незамедлительно обратиться к администратору СУБД. Механизмы восстановления СУБД указаны в документе «Руководство системного администратора».

4.4 Очистка данных в СУБД

Ниже приведены команды, которые можно использовать для очистки данных в СУБД.

- Vacuum – команда очищающая текущую базу данных, с которой работой пользователь;
- Vacuum <имя_объекта> - команда очищающая указанный объект;
- Vacuum full – команда очищающая текущую базу данных и переупорядочивающая блоки в сегментах данных;
- Cluster – синоним команды vacuum full.

Формат использования команд:

VACUUM [(параметр [, ...])] [таблица_и_столбцы [, ...]]

VACUUM [FULL] [FREEZE] [VERBOSE] [ANALYZE] [таблица_и_столбцы [, ...]]

где параметром может быть:

FULL [логическое_значение]

FREEZE [логическое_значение]

VERBOSE [логическое_значение]

ANALYZE [логическое_значение]

DISABLE_PAGE_SKIPPING [логическое_значение]

SKIP_LOCKED [логическое_значение]

INDEX_CLEANUP { AUTO | ON | OFF }

PROCESS_TOAST [логическое_значение]

TRUNCATE [логическое_значение]

PARALLEL целое_число

где *таблица_и_столбцы* это:

имя_таблицы [(имя_столбца [, ...])]

Все указанные команды принимают параметры:

FULL

Выбирает «полную» очистку, которая может освободить больше места, но занимает гораздо больше времени и ставит на таблицы исключительную блокировку. Кроме того, этот метод требует дополнительного дискового пространства, так как он записывает новую копию таблицы и не освобождает старую копию до завершения операции.

FREEZE

Выбирает принудительное «замораживание» (помещение строк за транзакционный горизонт событий) строк. Указание **FREEZE** равнозначно выполнению VACUUM с нулевыми значениями параметров `vacuum_freeze_min_age` и `vacuum_freeze_table_age`. При перезаписи таблицы принудительное замораживание выполняется всегда, поэтому при указании **FULL** этот параметр избыточен.

VERBOSE

Выводит подробный отчет о процедуре очистки для каждой таблицы.

ANALYZE

Обновляет статистику, используемую планировщиком для определения наиболее эффективного способа выполнения запроса.

DISABLE_PAGE_SKIPPING

Обычно VACUUM будет пропускать страницы, основываясь на карте видимости. Страницы, где все кортежи значатся замороженными, всегда можно пропускать, а те, где все кортежи будут видны для всех транзакций, можно обрабатывать, только когда выполняется агрессивная очистка. Кроме того, за тем же исключением, некоторые страницы можно пропускать, чтобы не ждать, когда их

закончат использовать другие сеансы. Этот параметр выключает поведение пропуска страниц и предназначен для использования, только когда содержимое карты видимости является подозрительным, а это должно происходить, только если существует аппаратная или программная проблема, вызывающая повреждение базы данных.

SKIP_LOCKED

Указывает, что в начале работы над отношением VACUUM не должен ждать освобождения конфликтующих блокировок: если отношение нельзя немедленно заблокировать без ожидания, оно пропускается. При использовании этого параметра VACUUM по-прежнему может блокироваться при открытии индексов отношения. VACUUM ANALYZE может блокироваться при получении выборочных строк из секций, потомков таблиц и некоторых типов сторонних таблиц. VACUUM обрабатывает все секции указанных таблиц, этот параметр позволит VACUUM пропустить их, если на таблице имеется конфликтующая блокировка.

INDEX_CLEANUP

VACUUM будет пропускать очистку индексов, когда в таблице имеется всего несколько неиспользуемых строк. Предполагается, что стоимость обработки всех индексов таблицы значительно превысит выгоду от удаления из индекса этих неиспользуемых строк. Этот параметр можно использовать, чтобы заставить VACUUM обрабатывать индексы, содержащие один и более неиспользуемых строк. При значении *AUTO* (по умолчанию) VACUUM может пропускать очистку индексов, когда это применимо. Если **INDEX_CLEANUP** установлен в значение *ON*, VACUUM будет консервативно удалять из индексов все неиспользуемые строки.

INDEX_CLEANUP можно установить в значение *OFF*, чтобы заставить VACUUM **всегда** пропускать очистку индексов, даже когда в таблице имеется много неиспользуемых кортежей. Используется при необходимости выполнить VACUUM как можно быстрее, чтобы избежать надвигающейся угрозы зацикливания идентификаторов транзакций. Механизм защиты от этого зацикливания, управляемый параметром `vacuum_failsafe_age`, обычно срабатывает автоматически, поэтому более предпочтителен. Если очистка индекса не проводится регулярно,

может пострадать производительность, поскольку при изменениях таблицы индексы будут накапливать неиспользуемые строки, а сама таблица будет накапливать неиспользуемые указатели строк, которые нельзя удалить до завершения очистки индекса.

Этот параметр не действует на таблицы, в которых нет индексов, и игнорируется при использовании параметра **FULL**. Также он не действует на механизм защиты от заикливания идентификаторов транзакций. При срабатывании он будет пропускать очистку индексов, даже если в **INDEX_CLEANUP** установлено **ON**.

PROCESS_TOAST

Указывает, что **VACUUM** должна попытаться обработать соответствующую таблицу **TOAST** для каждого отношения, если таковая существует. Обычно это желаемое поведение, которое установлено по умолчанию. Установка для этого параметра значения **false** может быть полезно, если нужно очистить только основное отношение. Этот параметр требуется при использовании параметра **FULL**.

TRUNCATE

Указывает, что **VACUUM** должна попытаться отрезать любые пустые страницы в конце таблицы, чтобы дисковое пространство из-под усеченных страниц могло быть возвращено операционной системе. Обычно это желаемое поведение, которое установлено по умолчанию, если только параметр **vacuum_truncate** очищаемой таблицы не был установлен в значение **false**. Установка этого параметра в **false** может быть полезна, чтобы избежать блокировки **ACCESS EXCLUSIVE** на таблицы, требующей усечения. Этот параметр игнорируется при использовании параметра **FULL**.

PARALLEL

Осуществляет фазы очистки и уборки индексов при параллельном выполнении **VACUUM**, используя **целое_число** фоновых рабочих процессов (подробную информацию о каждой фазе очистки см. в таблице Фазы **VACUUM**). Количество рабочих процессов, используемых для этой операции, равно количеству индексов в отношении, поддерживающих параллельную очистку. Это количество ограничено

количеством рабочих процессов, заданных параметром **PARALLEL** (если он указан), которое дополнительно ограничено параметром `max_parallel_maintenance_workers`. Индекс может участвовать в параллельной очистке только в том случае, если его размер больше, чем `min_parallel_index_scan_size`. Это не гарантирует, что при выполнении данной операции будет задействовано ровно столько параллельных рабочих процессов, сколько задано **целым_числом**. Возможно, что в процессе очистки рабочие процессы будут использоваться в меньшем количестве или не будут использоваться вовсе. Каждый индекс может обрабатывать только один рабочий процесс, поэтому параллельные рабочие процессы запускаются, только если в таблице имеется как минимум 2 индекса. Рабочие процессы очистки запускаются до начала каждой фазы и завершаются после ее окончания. В будущих релизах это поведение может измениться. Данный параметр нельзя использовать вместе с **FULL**.

логическое_значение

Указывает, должен ли выбранный параметр быть включен или выключен. Для включения этого параметра можно написать **TRUE**, **ON** или **1**, а для выключения — **FALSE**, **OFF** или **0**. Также **логическое_значение** можно опустить, и в этом случае предполагается **TRUE**.

целое_число

Указывает неотрицательное целое значение, передаваемое выбранному параметру.

имя_таблицы

Имя конкретной таблицы или материализованного представления, подлежащих очистке (может быть дополнено схемой). Если указанная таблица является разделенной на секции, все ее листовые секции тоже очищаются.

имя_столбца

Имя конкретного столбца для анализа. По умолчанию производится анализ всех столбцов. Если указан список столбцов, также должно быть указано **ANALYZE**.

Пример: очистка единственной таблицы `onek`, анализ ее для оптимизатора и выведение подробного отчета об операции очистки:

```
VACUUM (VERBOSE, ANALYZE) onek;
```

Информация для команды VACUUM ANALYZE доступна в представлении pg_statistic.

```
SELECT * from pg_statistic
```

Приведенная команда приведена в общем виде и может быть детализирована согласно правилам написания SQL запросов.

5 СООБЩЕНИЯ ПОЛЬЗОВАТЕЛЯМ

В СУБД «Квант-гибрид 1.5» сообщения пользователям предоставляются через сторонние клиентские приложения или через программную оболочку `psql` /командную строку.

Для работы с сообщениями СУБД клиентским приложениям используются коды ошибок, которые приведены в документе ВЕМР.00207-01 32 01 «Руководство системного администратора».

Сообщения об ошибках при подключении указаны в разделах «Сообщения об ошибках при аутентификации» и «Сообщения об ошибках при подключении клиентских приложений к СУБД» настоящего документа, описание полей сообщений с ошибками и замечаниями приведено в разделе ниже (раздел «Поля сообщений с ошибками и замечаниями»).

5.1 Поля сообщений с ошибками и замечаниями

В этом разделе описываются поля, которые могут содержаться в сообщениях **ErrorResponse** и **NoticeResponse**. У каждого типа поля имеется свой однобайтовый идентификационный маркер. Поле любого из этих типов должно содержаться в сообщении не более одного раза.

S

Важность: это поле содержит **ERROR**, **FATAL** или **PANIC** (в сообщении об ошибке), либо **WARNING**, **NOTICE**, **DEBUG**, **INFO** или **LOG** (в сообщении с замечанием), либо их переведенные значения (ОШИБКА, ФАТАЛЬНО, ПАНИКА, ПРЕДУПРЕЖДЕНИЕ, ЗАМЕЧАНИЕ, ОТЛАДКА, ИНФОРМАЦИЯ, СООБЩЕНИЕ соответственно). Присутствует всегда.

V

Важность: это поле содержит **ERROR**, **FATAL** или **PANIC** (в сообщении об ошибке), либо **WARNING**, **NOTICE**, **DEBUG**, **INFO** или **LOG** (в сообщении с замечанием). Оно подобно полю S, но его содержимое никогда не переводится. Присутствует всегда.

C

Код: код SQLSTATE выданной ошибки (код приведен в документе BEMP.00207-01 32 01 «Руководство системного администратора»). Не переводится. Присутствует всегда.

M

Сообщение: основное сообщение об ошибке, удобное для восприятия пользователем. Оно точное, но краткое (обычно в одну строку). Присутствует всегда.

D

Детализация: необязательное дополнительное сообщение об ошибке, содержащее более детальную информацию о проблеме. Может состоять из нескольких строк.

H

Подсказка: необязательное предложение решения проблемы. Отличается от сообщения **Detail** тем, что предлагает совет, а не голые факты. Может состоять из нескольких строк.

P

Позиция: значение этого поля является целочисленным числом в ASCII, указывающим на положение курсора ошибки как на индекс в исходной строке запроса. Индекс первого символа — 1, при этом позиции отсчитываются по символам, а не по байтам.

p

Внутренняя позиция: она определяется так же, как поле **P**, но применяется, когда положение курсора относится ко внутренне сгенерированной команде, а не к команде, отправленной клиентом. Вместе с этим полем всегда присутствует поле **q**.

q

Внутренний запрос: текст внутренне сгенерированной команды, в которой произошла ошибка.

W

Где: указание на контекст, в котором произошла ошибка. Включает обратную трассировку стека текущих функций на процедурном языке и внутренне

сгенерированных запросов. Записи трассировки разделяются по строкам, вначале идет последняя.

s

Имя схемы: если ошибка связана с определенным объектом базы данных, в этом поле указывается имя схемы, содержащей этот объект, если таковая имеется.

t

Имя таблицы: если ошибка связана с определенной таблицей, в этом поле указывается имя этой таблицы.

c

Имя столбца: если ошибка связана с определенным столбцом таблицы, в этом поле указывается имя этого столбца.

d

Имя типа данных: если ошибка связана с определенным типом данных, в этом поле указывается имя этого типа данных.

n

Имя ограничения: если ошибка связана с определенным ограничением, в этом поле указывается имя этого ограничения. Узнать, к какой таблице или домену оно относится, можно, обратившись к полям, описанным выше. Индексы считаются ограничениями, даже если они были созданы не с синтаксисом ограничений.

F

Файл: имя файла в исходном коде, в котором была обнаружена ошибка.

L

Строка: номер строки в исходном коде, в которой была обнаружена ошибка.

R

Программа: имя программы в исходном коде, в которой была обнаружена ошибка.

СОКРАЩЕНИЯ И ОПРЕДЕЛЕНИЯ

Сокращение, обозначение	Расшифровка
Linux	Семейство Unix-подобных операционных систем на базе ядра Linux
psql	Пользовательская интерактивная утилита для работы из командной строки
QNB	Объектно-реляционная система управления базами данных «Квант-Гибрид 1.5»
QSS	«Quantum Secure Storage» (QSS), сертифицированный ФСБ России комплекс для криптографической защиты конфиденциальности и целостности информации «Quantum Secure Storage» (сертификат СФ/124-4721 от 15.01.2024). В Системе подключен в виде одноименного модуля.
Атрибут	Элемент с определенным именем и типом данных, содержащийся в кортеже
Б (байт), Мб (мегабайт), ГБ (гигабайт)	Байт (англ. byte) (русское обозначение: байт и Б; международное: В, byte) — единица хранения и обработки цифровой информации; совокупность битов, обрабатываемая компьютером одновременно 1 мегабайт = 1024 байта 1 гигабайт = 1024 мегабайта
База данных	Совокупность данных, которая рассматривается как модуль; основной целью базы данных является хранение и поиск связанной информации
БД	База данных

Сокращение, обозначение	Расшифровка
Клиент (процесс)	Любой процесс, возможно, удаленный, который запускает сеанс путем подключения к экземпляру СУБД для взаимодействия с базой данных
Кортеж	Упорядоченный набор атрибутов. Порядок атрибутов может определяться таблицей (или другим отношением), в которой содержится кортеж. В этом случае кортеж называют <i>строкой</i> таблицы. Он также может определяться структурой результирующего множества; такие кортежи иногда называют <i>записями</i>
ОС	Операционная система
Отношение	Общий термин, который охватывает все объекты в базе данных, имеющие имя и упорядоченный список атрибутов
Очистка (VACUUM)	Процесс удаления устаревших версий кортежей из таблиц или материализованных представлений и другая тесно связанная с ним обработка данных
Пользователь базы данных	Пользователь, который взаимодействует с СУБД и выполняет операции на объектах, хранимых в пределах базы данных
ПЭВМ	Персональная электронная вычислительная машина
Роль	Набор прав доступа к объектам экземпляра СУБД. Принадлежность к роли тоже может считаться правом, которым можно наделять другие роли
СУБД	Система управления базами данных
ЭВМ	Электронная вычислительная машина

