

**СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ «КАТРАПС» (СУБД  
«КАТРАПС») ВЕРСИИ 1.10.11**

---

РУКОВОДСТВО АДМИНИСТРАТОРА СУБД

Листов 22

МОСКВА

2025

## **Оглавление**

<b>Назначение и область применения СУБД «КАТРАПС».....</b>	<b>3</b>
<b>Общие сведения.....</b>	<b>3</b>
<b>Архитектура СУБД «КАТРАПС».....</b>	<b>4</b>
<b>Принципы безопасной работы СУБД «КАТРАПС».....</b>	<b>5</b>
<b>Механизм хранения по умолчанию.....</b>	<b>5</b>
<b>Структуры данных.....</b>	<b>6</b>
<b>Смена пароля Администратора СУБД (root).....</b>	<b>6</b>
<b>Доступ по сети.....</b>	<b>7</b>
<b>Настройка сервера для работы по сети.....</b>	<b>7</b>
<b>Настройка брандмауэра.....</b>	<b>8</b>
<b>Журналы в СУБД «КАТРАПС».....</b>	<b>8</b>
<b>События безопасности, связанные с доступными пользователям функциями.....</b>	<b>8</b>
<b>Режимы работы СУБД «КАТРАПС».....</b>	<b>9</b>
<b>Регистрируемые события.....</b>	<b>10</b>
<b>Матрица доступа СУБД «КАТРАПС».....</b>	<b>11</b>
<b>Управление ролями СУБД «КАТРАПС».....</b>	<b>14</b>
<b>Создание роли.....</b>	<b>14</b>
<b>Назначение роли.....</b>	<b>15</b>
<b>Удаление роли.....</b>	<b>15</b>
<b>Просмотр прав.....</b>	<b>16</b>

Администрирование демона `mariadb` .....	16
Действия после сбоев и ошибок эксплуатации.....	17
Приложение 1. Параметры команды mariadb-admin.....	18
Приложение 2. Системные привилегии СУБД «КАТРАПС».....	21

## Назначение и область применения СУБД «КАТРАПС»

### Общие сведения

СУБД «КАТРАПС» является программным средством, реализующим функциональные возможности по созданию баз данных, манипулированию данными (вставке, обновлению, удалению, выборке), обеспечению безопасности, надежности хранения и целостности данных, администрированию баз данных, а также обеспечивающим управление доступом субъектов доступа к объектам доступа баз данных, предназначенных для хранения информации, подлежащей защите в информационной (автоматизированной) системе. СУБД «КАТРАПС» должна обеспечивать контроль и управление данными, позволяя выполнять различные административные операции, такие как мониторинг производительности, настройка, а также резервное копирование и восстановление.

Основными задачами, решаемыми СУБД «КАТРАПС», являются:

- централизованное хранение данных;
- управление данными;
- обработка данных;
- оптимизация запросов;
- поддержка языков баз данных;
- обеспечение удобства работы с данными;
- маскирование самих табличных, а также внешне сохраняемых ссылочных текстовых и графических данных перед их удалением.

СУБД «КАТРАПС» функционирует в среде следующих операционных систем:

- Альт 8 СП (сертификат соответствия ФСТЭК России № 3866, выдан 10.08.2018 г., действителен до 10.08.2028 г.);
- РЕД ОС (сертификат соответствия ФСТЭК России № 4060, выдан 12.01.2019 г., действителен до 12.01.2029 г.).

СУБД «КАТРАПС» реализует следующие меры защиты в соответствии с Требованиями по безопасности информации к системам управления базами данных, утвержденными приказом ФСТЭК России от 14 апреля 2023 г. № 64:

- Управление доступом в системе управления базами данных;
- Идентификация и аутентификация пользователей в системе управления базами данных;
- Контроль целостности в системе управления базами данных;
- Регистрация событий безопасности в системе управления базами данных;

- Резервное копирование и восстановление в системе управления базами данных;
- Обеспечение доступности в системе управления базами данных;
- Очистка памяти в системе управления базами данных;
- Ограничение программной среды в системе управления базами данных;
- Управление параметрами производительности системы управления базами данных.

## Архитектура СУБД «КАТРАПС»

Следующая схема представляет архитектуру СУБД «КАТРАПС»:

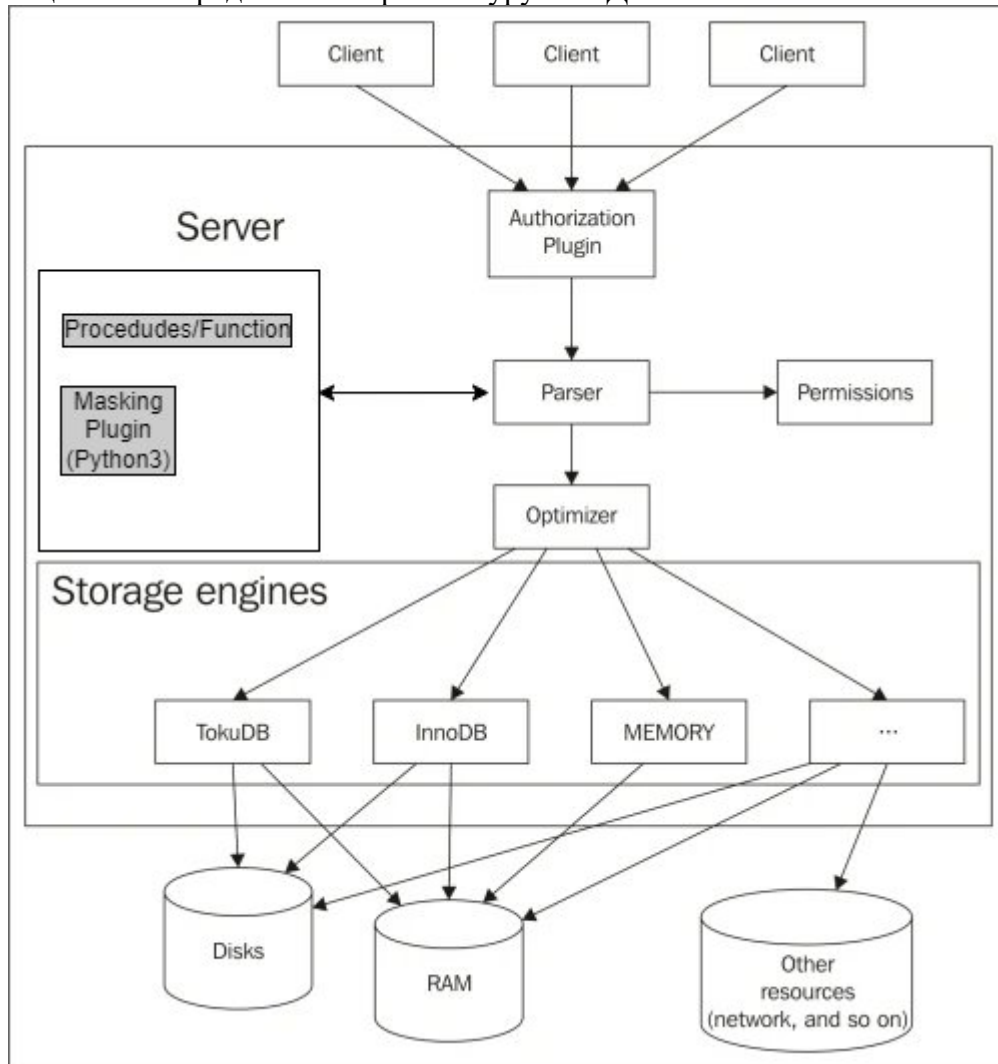


Рисунок 1. Архитектура СУБД «КАТРАПС»

С точки зрения конечного пользователя, СУБД «КАТРАПС» получает некоторые SQL-запросы или операторы, уточняет их и возвращает набор результатов. Этот процесс включает следующие шаги:

- Когда клиент подключается к СУБД «КАТРАПС», аутентификация выполняется на основе имени хоста, имени пользователя и пароля клиента. Аутентификацию можно дополнительно делегировать плагину.
- Если вход успешен, клиент может отправить SQL-запрос на сервер.
- Анализатор понимает строку SQL.
- Сервер проверяет, имеет ли клиент разрешения, необходимые для запрошенного действия.

- Если запрос хранится в кэше запросов, результаты немедленно возвращаются клиенту.
- Оптимизатор попытается найти самую быструю стратегию выполнения или план запроса. Это означает, что оптимизатор определяет порядок чтения таблиц. Он также решает, к каким индексам будет осуществляться доступ и будет ли использоваться временная внутренняя таблица. Хорошая стратегия может значительно сократить доступ к дискам и на порядок снизить сложность операций.
- Механизмы хранения читают и записывают файлы данных и индексов, а также любой кэш, который они могут использовать для ускорения операций. Некоторые важные функции, такие как транзакции и внешние ключи, реализованы на уровне механизма хранения.

СУБД «КАТРАПС» и механизмы хранения ведут набор журналов для отслеживания полученных операторов, возникших ошибок, изменений в данных и т. д. Большинство журналов не являются обязательными; однако некоторые журналы необходимы для некоторых административных задач. Например, двоичный журнал позволяет выполнять резервное копирование или репликацию.

СУБД «КАТРАПС» имеет несколько опций, влияющих на поведение сервера. Многие из них являются динамическими, что означает, что их можно изменять во время выполнения; другие являются статическими, что означает, что значение, назначенное во время запуска сервера, не может измениться. Большинство из них существуют как на уровне сеанса, что означает, что любой отдельный пользователь может изменить значение текущего соединения, так и на глобальном уровне, который применяется ко всем пользователям, которые не установили значение сеанса. Параметр можно указать несколькими способами, например, в параметрах командной строки сервера, в файлах конфигурации или, если он динамический, с помощью оператора SQL. СУБД «КАТРАПС» читает набор файлов конфигурации в заданном порядке. Файл конфигурации находится в каталоге установки СУБД «КАТРАПС».

## Принципы безопасной работы СУБД «КАТРАПС»

Безопасная работа СУБД «КАТРАПС» основана на механизмах безопасности:

- Управления доступом;
- Идентификации и аутентификации;
- Контроля целостности;
- Регистрации событий безопасности;
- Резервного копирования и восстановления;
- Обеспечения доступности;
- Очистки памяти;
- Ограничения программной среды;
- Управления параметрами производительности.

Механизмы безопасности СУБД «КАТРАПС» реализованы посредством обеспечения мер защиты, реализующих функции безопасности, применяемые в соответствии с «Требованиями по безопасности информации к системам управления базами данных», утвержденными приказом ФСТЭК России от 14 апреля 2023 г. № 64.

## Механизм хранения по умолчанию

InnoDB является механизмом хранения по умолчанию. В СУБД «КАТРАПС» используется его обновленная версия XtraDB - это InnoDB с исправлениями ошибок и некоторыми уникальными функциями (в основном для производительности и мониторинга).

InnoDB (XtraDB) — это высокопроизводительный механизм хранения общего назначения, который поддерживает транзакции с точками сохранения, транзакции ХА и внешние ключи. Точки сохранения — это промежуточные состояния, которые можно сохранить в середине транзакции

и затем при необходимости восстановить. ХА — это особый тип транзакции, предназначенный для операций, в которых задействовано несколько ресурсов, не обязательно баз данных SQL. В большинстве случаев производительность InnoDB лучше, чем у других механизмов.

InnoDB транзакции реализуются через сложную систему блокировки и журналы отмены. Каждая блокировка включает в себя одну строку или диапазон строк; строки идентифицируются с помощью индексных записей. Журналы отмены используются для отката транзакций при необходимости и могут храниться в системном табличном пространстве или где-либо еще.

## Структуры данных

В СУБД «КАТРАПС» по умолчанию InnoDB отображается в XtraDB, совместимую версию InnoDB.

Таблицы InnoDB содержатся в **табличных пространствах**. Табличное пространство — это файл, содержащий данные и индексы для одной или нескольких таблиц. Если для системной переменной `innodb_file_per_table` установлено значение 1, что является значением по умолчанию, каждая таблица хранится в отдельном табличном пространстве. Эта переменная является динамической, поэтому некоторые таблицы можно хранить в отдельных файлах, а другие — в системном табличном пространстве.

Системное табличное пространство по умолчанию также содержит словарь данных InnoDB, журналы отмены, буфер изменений и буфер двойной записи. Словарь данных представляет собой коллекцию метаданных всех таблиц, столбцов и индексов InnoDB. Системное табличное пространство хранится в каталоге `data`, в файлах `ibdata` (по умолчанию два файла).

Часть табличного пространства называется **сегментом**. Обычные табличные пространства имеют один сегмент для данных и один сегмент для каждого индекса. Системное табличное пространство состоит из нескольких сегментов.

**Страница** — это небольшая единица данных, хранящаяся в табличном пространстве или в пуле буферов. Страницы имеют фиксированный размер, который можно настроить. Страница содержит одну или две строки и обычно некоторое пустое пространство. Коэффициент непустого пространства называется **коэффициентом заполнения**.

Страница, которая была изменена в буфере изменений, называется **грязной страницей**.

В некоторых случаях, например, для согласованного процесса чтения, InnoDB последовательно читает несколько страниц вместе общим размером 1 МБ. Такие группы страниц называются **экстендами**.

Индексы InnoDB важны не только для чтения, но и для блокировок. Каждая блокировка указывает на индексную запись.

Индекс InnoDB может быть кластерным индексом или вторичным индексом. Первичные ключи представляют собой кластерные индексы. Если у таблицы нет первичного ключа, в качестве первичного ключа будет использоваться первый UNIQUE индекс, содержащий только NOT NULL столбцы. Если такого индекса не существует, автоматически создается скрытый индекс кластера. Все записи вторичного индекса указывают на запись кластеризованного индекса, поэтому можно сказать, что все вторичные индексы содержат кластеризованный индекс.

## Смена пароля Администратора СУБД (root)

После первого запуска сервера необходимо задать пароль для Администратора СУБД «КАТРАПС» следующим образом:

Запустите сервер:

```
> sudo systemctl start mariadb
```

Соединитесь с базой данных как пользователь **root**, без пароля:

```
> sudo mysql -u root
```

Измените пароль **root**:

```
> ALTER USER 'root'@'localhost' IDENTIFIED BY '<new_password>';  
  
FLUSH PRIVILEGES;
```

Обновите конфигурацию **systemd**:

```
> sudo systemctl daemon-reload
```

Проверьте, что сервер работает корректно:

```
> sudo systemctl status mariadb
```

Остановите сервер при возникшей необходимости:

```
> sudo systemctl stop mariadb
```

## Доступ по сети

Для успешного подключения к СУБД «КАТРАПС» по сети нужно выполнить 3 условия:

1. Создать правильную учетную запись.
2. Сервер баз данных должен слушать сетевые запросы.
3. Правила брандмауэра не должны блокировать доступ по порту **mysql** (по умолчанию **3306**).

### Настройка сервера для работы по сети

Проверить, на каком сетевом интерфейсе слушает сервер можно командой:

```
> ss -tunlp | grep 3306
```

Если запросы выполняются только на локальных адресах:

```
> tcp LISTEN 0 50 127.0.0.1:3306 ...
```

*\* сервер слушает на адресе **127.0.0.1**, что означает обработку только локальных запросов.*

То необходимо настроить сервер с использованием конфигурационного файла, находящегося по адресу:

```
> vi /etc/my.cnf.d/mariadb-server.cnf.
```

Необходимо значение для опции **bind-address**:

```
> bind-address = 0.0.0.0
```

*\* в данном примере серверу разрешаем слушать на любом адресе (**0.0.0.0**). Если нужно ограничить сетку прослушивания, надо вписать адрес доступного в сети шлюза.*

Перезапуск сервиса:

```
> systemctl restart mysql
```

## Настройка брандмауэра

а) Для iptables (как правило, в системах на основе deb):

```
> iptables -I INPUT -p tcp --dport 3306 -j ACCEPT
```

Чтобы сохранить правила, можно использовать iptables-persistent:

```
> apt install iptables-persistent
```

```
> netfilter-persistent save
```

б) Для firewalld (как правило, в системах на основе rpm):

```
> firewall-cmd --permanent --add-port=3306/tcp
```

```
> firewall-cmd --reload
```

## Журналы в СУБД «КАТРАПС»

СУБД «КАТРАПС» ведет следующие журналы:

- **Error log:** Это журнал содержит ошибки, произошедшие во время выполнения сервера. Сюда входят как проблемы с сервером (например, ошибки, препятствующие запуску плагинов), так и ошибки SQL.
- **SQL\_ERROR\_LOG:** журнал записывает ошибки, сгенерированные операторами SQL, в файл. Это более конкретный вариант, чем журнал ошибок, поскольку в нем регистрируются только ошибки SQL.
- **General query log:** журнал, содержащий операторы SQL.
- **Slow query logs:** журнал для хранения запросов, которые занимают больше заданного времени или не используют какой-либо индекс. Это полезно для выяснения, почему приложение или база данных работает медленно.
- **Binary log (binlog):** в зависимости от выбранного формата этот журнал содержит данные, измененные в двоичную форму, или инструкции SQL, вызвавшие изменение. Это необходимо для реализации инкрементного резервного копирования, репликации или кластера базы данных.
- **Relay log:** Этот журнал существует только на ведомых устройствах репликации и содержит данные, полученные ведущим устройством. Каждая запись в журнале ведомого устройства соответствует записи в двоичном журнале ведущего устройства.

## События безопасности, связанные с доступными пользователям функциями

В СУБД «КАТРАПС» осуществляется:

- регистрация событий безопасности, связанных с функционированием СУБД «КАТРАПС» и действиями пользователей СУБД «КАТРАПС»;
- оповещение администратора системы управления базами данных, администратора базы данных (администратора информационной системы) о событиях безопасности;
- сбор и хранение записей в журнале событий безопасности, которые позволяют определить, когда и какие события происходили.

Регистрации подлежат следующие события безопасности:

- создание учетных записей пользователей СУБД «КАТРАПС»;
- изменение атрибутов учетных записей пользователей СУБД «КАТРАПС»;

- успешные и неуспешные попытки аутентификации пользователей СУБД «КАТРАПС»;
- запуск и остановка СУБД «КАТРАПС» с указанием причины остановки;
- изменение конфигурации СУБД «КАТРАПС»;
- создание и удаление базы данных, таблицы за исключением временных таблиц, создаваемых СУБД «КАТРАПС» в служебных целях;
- подключение, восстановление базы данных;
- изменение правил разграничения доступа в СУБД «КАТРАПС»;
- факты нарушения целостности объектов контроля;
- создание и изменение процедур (программного кода), хранимых в базах данных, и представлений.

Для регистрируемых событий безопасности в каждой записи журнала событий безопасности регистрируются:

- номер (уникальный идентификатор) события;
- дата, время события безопасности;
- тип события безопасности;
- сведения о важности события.

Записи журнала событий безопасности представляются в структурированном виде и содержат дату и время события безопасности, взятое из аппаратной платформы или операционной системы.

Журнал событий безопасности СУБД «КАТРАПС» доступен для чтения администратору СУБД и администратору ИС. Для пользователя ИС журнал событий безопасности СУБД «КАТРАПС» недоступен.

При исчерпании области памяти, отведенной под журнал событий безопасности, СУБД «КАТРАПС» самостоятельно осуществляет архивирование журнала с последующей очисткой высвобождаемой области памяти.

## **Режимы работы СУБД «КАТРАПС»**

Для СУБД «КАТРАПС» определены следующие режимы функционирования:

- штатный (основной) режим;
- режим восстановления.

Штатный режим является основным режимом функционирования в соответствии с показателями назначения, обеспечивающим выполнение полного объема функций СУБД «КАТРАПС».

Режим восстановления — это режим, используемый для восстановления после чрезвычайных ситуаций. Прежде чем вносить изменения, необходимо убедиться, что имеется резервная копия базы данных на случай, если понадобится ее восстановить. Системная переменная сервера `innodb_force_recovery` задает режим восстановления. Режим 0 соответствует нормальному использованию, а чем выше режим, тем более строгие ограничения. Высшие режимы включают в себя все ограничения низших режимов.

Для режима восстановления никогда не следует устанавливать значение, отличное от нуля, за исключением экстренных ситуаций.

Режим восстановления не устраняет повреждения. Поврежденные файлы остаются поврежденными независимо от режима восстановления. Единственная цель режима восстановления — предоставить доступ для чтения к данным, если это вообще возможно.

Как правило, лучше начинать с режима восстановления 1 и при необходимости увеличивать его с небольшим шагом. При режиме восстановления < 4 должны быть потеряны только поврежденные страницы. При значении 4 вторичные индексы могут быть повреждены. При значении 5 результаты могут быть противоречивыми, а вторичные индексы могут быть повреждены (даже если при значении 4 это не так). Значение 6 оставляет страницы в устаревшем состоянии, что может привести к еще большему повреждению.

Доступны следующие режимы восстановления:

Режим	Описание
0	Режим по умолчанию, когда СУБД работает нормально. Транзакции записи разрешены с помощью <code>innodb force recovery&lt;=4</code> .
1	( <code>SRV_FORCE_IGNORE_CORRUPT</code> ) позволяет серверу продолжать работу даже в случае обнаружения поврежденных страниц. Это достигается за счет того, что при восстановлении на основе журнала повторов игнорируются определенные ошибки, такие как отсутствующие файлы данных или поврежденные страницы данных. Любой журнал повторов для затронутых файлов или страниц будет пропущен. Можно облегчить сброс таблиц, заставив оператор <code>SELECT * FROM table_name</code> перепрыгивать через поврежденные индексы и страницы.
2	( <code>SRV_FORCE_NO_BACKGROUND</code> ) останавливает работу главного потока, предотвращая сбой, возникающий во время очистки. Очистка выполняться не будет, поэтому журналы отмены будут продолжать расти.
3	( <code>SRV_FORCE_NO_TRX_UNDO</code> ) не откатывает транзакции DML после аварийного восстановления. Не влияет на откат текущих активных транзакций DML. Также будет предотвращен запуск некоторых фоновых задач, генерирующих отмену. Эти задачи могут попасть в ожидание блокировки из-за восстановленных незавершенных транзакций, откат которых предотвращается.
4	( <code>SRV_FORCE_NO_DDL_UNDO</code> ) не откатывает транзакции после аварийного восстановления. Не влияет на откат активных в данный момент транзакций. Также будет предотвращен запуск некоторых фоновых задач, генерирующих отмену. Эти задачи могут попасть в ожидание блокировки из-за восстановленных незавершенных транзакций, откат которых предотвращается.
5	( <code>SRV_FORCE_NO_UNDO_LOG_SCAN</code> ) рассматривает незавершенные транзакции как зафиксированные и не просматривает <code>undo logs</code> при запуске. Любой журнал DDL для таблиц будет по существу игнорироваться, но сервер запустится.
6	( <code>SRV_FORCE_NO_LOG_REDO</code> ) не выполняет повтор транзакций журнала повторов в рамках восстановления. В этом активном режиме выполнение запросов, требующих индексов, скорее всего, завершится неудачно. Однако, если дампы таблицы по-прежнему вызывает сбой, можно попробовать использовать <code>SELECT * FROM tab ORDER BY primary_key DESC</code> для дампа всей части данных после поврежденной части.

## Регистрируемые события

Доступ к механизму регистрации событий в СУБД «КАТРАПС» имеет только пользователь с ролью администратора СУБД. Аудит событий обеспечивается плагином `server_audit.so`. Список переменных, связанных с плагином аудита, приведен в Таблица 1.

Таблица 1. Ролевая модель СУБД «КАТРАПС»

Переменная	Описание	Значение
<code>server_audit_events</code>	ограничивает ведение журнала аудита определенными типами событий	CONNECT, QUERY, TABLE, QUERY_DDL, QUERY_DML, QUERY_DCL, QUERY_DML_NO_SELECT
<code>server_audit_excl_users</code>	содержит список пользователей, активность которых не будет регистрироваться	Строковое значение до 1024 символов

server_audit_file_path	задает путь и имя файла журнала	server_audit.log
server_audit_file_rotate_now	запуск принудительной ротации файла журнала	OFF/ ON
server_audit_file_rotate_size	размер файла журнала (в байтах), при достижении которого выполняется ротация	От 100 до 9223372036854775807
server_audit_file_rotations	количество ротаций, которые нужно сохранить	От 0 до 999
server_audit_incl_users	содержит список пользователей, чья активность будет регистрироваться	Строковое значение до 1024 символов
server_audit_loc_info	используется внутренними компонентами плагина, скрыт для пользователя	Строковое значение до 1024 символов
server_audit_logging	включает/отключает ведение журнала	OFF/ ON
server_audit_mode	содержит версию сервера, с которой был запущен плагин	От 0 до 1
server_audit_output_type	желаемый тип вывода	Syslog, file
server_audit_query_log_limit	ограничение длины строки запроса в записи	от 0 до 2147483647
server_audit_syslog_facility	параметры записей, которые будут отправлены в системный журнал, журнал можно отфильтровать по этому параметру	LOG_USER, LOG_MAIL, LOG_DAEMON, LOG_AUTH, LOG_SYSLOG, LOG_LPR, LOG_NEWS, LOG_UUCP, LOG_CRON, LOG_AUTHPRIV, LOG_FTP, and LOG_LOCAL0–LOG_LOCAL7
server_audit_syslog_ident	строковое значение для части «ident» каждой записи системного журнала	mysql-server_auditing
server_audit_syslog_info	строка «info», которая будет добавлена в записи системного журнала	Строковое значение до 1024 символов
server_audit_syslog_priority	определяет приоритет записей журнала для syslogd	LOG_EMERG, LOG_ALERT, LOG_CRIT, LOG_ERR, LOG_WARNING, LOG_NOTICE, LOG_INFO, LOG_DEBUG

## Матрица доступа СУБД «КАТРАПС»

В рамках ролевого метода управления доступом в СУБД «КАТРАПС» предусмотрены следующие роли:

- Администратор СУБД;
- Администратор ИС;
- Пользователь ИС.

Пользователям в рамках назначенных ролей доступны действия, приведенные в Таблице 2. Ролевая модель СУБД «КАТРАПС»

Роль	Объект доступа	Право (тип доступа)
Администратор СУБД	СЕРВЕР БАЗЫ ДАННЫХ	Инициализация СУБД
	СЕРВЕР БАЗЫ ДАННЫХ	Остановка СУБД
	СЕРВЕР БАЗЫ ДАННЫХ	Проведение unit-тестирования
	КОНФИГУРАЦИЯ СУБД	Изменение конфигурации СУБД
	КЭШ	Обновление кэша сервера
	БАЗА ДАННЫХ	Создание баз данных уровня ИС (CREATE DATABASE)
	БАЗА ДАННЫХ	Удаление баз данных уровня ИС (DROP DATABASE)
	ТАБЛИЦА РАЗРЕШЕНИЙ	Перезагрузка таблиц разрешений (RELOAD)
	КЭШ ТАБЛИЦЫ	Обновление кэша таблиц (RELOAD)
	ФАЙЛ	Чтение файлов, хранимых на сервере
	ФАЙЛ	Запись файлов, хранимых на сервере
	ФАЙЛ ОБНОВЛЕНИЯ	Установка обновлений
	ПРОЦЕСС	Просмотр информации о внутренних потоках сервера
	ПРОЦЕСС	Удаление информации о внутренних потоках сервера
	ПОЛЬЗОВАТЕЛЬ	Создание администраторов ИС (CREATE USER)
	ПОЛЬЗОВАТЕЛЬ	Удаление администраторов ИС (DELETE USER)
	ПОЛЬЗОВАТЕЛЬ	Изменение администраторов ИС (ALTER USER)
	ПОЛЬЗОВАТЕЛЬ	Блокирование администраторов ИС
	ПОЛЬЗОВАТЕЛЬ	Разблокирование администраторов ИС
	ЖУРНАЛ	Обновление журналов (RELOAD)
РЕЗЕРВНАЯ КОПИЯ	Создание резервной копии (резервирование данных)	
РЕЗЕРВНАЯ КОПИЯ	Загрузка резервной копии (восстановление данных)	
Администратор ИС	КОНФИГУРАЦИЯ БД	Изменение конфигурации ИС
	БАЗА ДАННЫХ	Создание баз данных уровня ИС (CREATE DATABASE)
	БАЗА ДАННЫХ	Удаление баз данных уровня ИС (DROP DATABASE)
	БАЗА ДАННЫХ	Изменение баз данных уровня ИС (ALTER DATABASE)
	ТАБЛИЦА	Создание таблиц (CREATE TABLE)
	ТАБЛИЦА	Удаление таблиц (DROP TABLE)
	ТАБЛИЦА	Изменение таблиц (ALTER TABLE)
	ТАБЛИЦА	Затирание (перезапись) битовой последовательностью (запуск python-скрипта в консоли)
	ИНДЕКС	Изменение индексов (ALTER INDEX)
	ПРОЦЕДУРА	Загрузка хранимых процедур (CREATE PROCEDURE)
	ФУНКЦИЯ	Загрузка функций (CREATE FUNCTION)
	ПОЛЬЗОВАТЕЛЬ	Создание пользователей ИС (CREATE USER)

	ПОЛЬЗОВАТЕЛЬ	Удаление пользователей ИС (DELETE USER)
	ПОЛЬЗОВАТЕЛЬ	Изменение пользователей ИС (ALTER USER)
	ПОЛЬЗОВАТЕЛЬ	Изменение прав пользователей ИС (GRANT)
	ПОЛЬЗОВАТЕЛЬ	Изменение привилегий доступа (роли) пользователей ИС (SET ROLE)
	ПОЛЬЗОВАТЕЛЬ	Блокирование пользователей ИС
	ПОЛЬЗОВАТЕЛЬ	Разблокирование пользователей ИС
	РОЛЬ	Создание роли
	РОЛЬ	Изменение роли
	РОЛЬ	Удаление роли
	ФАЙЛ	Чтение файлов, хранимых на сервере
	ФАЙЛ	Запись файлов, хранимых на сервере
	ПРОЦЕСС	Просмотр информации о внутренних потоках сервера
	ПРОЦЕСС	Удаление информации о внутренних потоках сервера
	ПРОЦЕДУРА	Запуск процедур (программного кода), хранимых в ИС
	ЖУРНАЛ	Просмотр сеансов доступа и выполненных пользователем действий
	РЕЗЕРВНАЯ КОПИЯ	Создание резервной копии ИС (резервирование данных)
	РЕЗЕРВНАЯ КОПИЯ	Загрузка резервной копии ИС (восстановление данных)
Пользователь ИС	ТАБЛИЦА	Создание таблиц (CREATE TABLE)
	ТАБЛИЦА	Удаление таблиц (DROP TABLE)
	ТАБЛИЦА	Изменение таблиц (ALTER TABLE)
	ТАБЛИЦА	Вставка новых записей в таблицы (INSERT INTO TABLE)
	ТАБЛИЦА	Извлечение существующих записей из таблиц (SELECT FROM TABLE)
	ТАБЛИЦА	Изменение существующих записей таблиц (UPDATE TABLE)
	ТАБЛИЦА	Удаление существующих записей из таблицы (DELETE FROM TABLE)
	ИНДЕКС	Создание индексов (CREATE INDEX)
	ИНДЕКС	Удаление индексов (DROP INDEX)
	ИНДЕКС	Изменение индексов (ALTER INDEX)
	ФАЙЛ	Чтение файлов, хранимых на сервере
	ФАЙЛ	Запись файлов на сервер
	ПРОЦЕДУРА	Запуск доступной хранимой процедуры в целях выполнения программного кода ИС

Дискреционный метод управления доступом субъектов доступа к объектам доступа СУБД «КАТРАПС» (база данных, таблица, запись или столбец, поле, представление, процедура (программный код) и т.д.) осуществляется на основе настраиваемых списков управления доступом (матриц управления доступом).

Списки управления доступом (матрицы управления доступом) позволяют задавать разрешение или запрет пользователям СУБД «КАТРАПС» создавать, изменять, удалять, исполнять процедуры (программный код), хранимые в базе данных.

Списки управления доступом (матрицы управления доступом) позволяют задавать разрешение или запрет пользователям и процедурам (программному коду), хранимым в базе данных, создавать, изменять, удалять, читать базы данных, таблицы, записи и иные объекты доступа.

Ролевая модель СУБД «КАТРАПС» предусматривает создание информационных систем (ИС) посредством создания «Администраторов ИС». «Администратор ИС» может создавать свои схемы баз данных, таблицы, индексы, временные таблицы, а также использовать модуль маскирования остаточной информации, работоспособность которого описана в отдельном руководстве. Задачами же «Пользователя ИС» являются работа непосредственно в доступной для него ИС, выполнение действий по наполнению данными таблиц, редактированию их и осуществлением необходимых выборок данных согласно наделяемых им полномочиям.

## Управление ролями СУБД «КАТРАПС»

Роль объединяет ряд привилегий. Он помогает более крупным организациям, где обычно несколько пользователей имеют одинаковые привилегии.

### Создание роли

Роли создаются с помощью инструкции CREATE ROLE и удаляются с помощью инструкции DROP ROLE. Затем роли назначаются пользователю с помощью расширения оператора GRANT, а привилегии назначаются роли обычным способом с помощью GRANT. Аналогично, оператор REVOKE может использоваться как для отзыва роли у пользователя, так и для отзыва привилегий у роли.

Оператор CREATE ROLE создает одну или несколько ролей. Чтобы использовать его, у пользователя должна быть глобальная привилегия CREATE USER или привилегия INSERT для базы данных mysql. Для каждой учетной записи создается новая строка в таблице mysql.user, не имеющая привилегий и с соответствующим полем is\_role, установленным в значение Y. Он также создает запись в таблице mysql.roles\_mapping.

Синтаксис создания роли:

```
CREATE [OR REPLACE] ROLE [IF NOT EXISTS] role

[WITH ADMIN

{CURRENT_USER | CURRENT_ROLE | user | role}]
```

Максимальная длина роли — 128 символов. Имена ролей можно заключать в кавычки.

PUBLIC и NONE зарезервированы и не могут использоваться в качестве имен ролей. NONE используется для отмены роли, PUBLIC имеет специальное применение в других системах, поэтому зарезервировано для целей совместимости.

Необязательное предложение WITH ADMIN определяет, будет ли текущий пользователь, текущая роль или другой пользователь или роль использовать вновь созданную роль. Если это предложение опущено, WITH ADMIN CURRENT\_USER рассматривается как значение по умолчанию, что означает, что текущий пользователь сможет назначать эту роль другим пользователям.

Если используется необязательное предложение OR REPLACE, оно действует так:

```
DROP ROLE IF EXISTS name;

CREATE ROLE name ...;
```

При использовании предложения IF NOT EXISTS СУБД «КАТРАПС» вместо ошибки вернет предупреждение, если указанная роль уже существует. Не может использоваться вместе с предложением OR REPLACE.

## Назначение роли

После подключения пользователь может получить все привилегии, связанные с ролью, установив роль с помощью инструкции SET ROLE.

Синтаксис назначения роли:

```
> SET ROLE { role | NONE }
```

Оператор SET ROLE включает роль вместе со всеми связанными с ней разрешениями для текущего сеанса. Чтобы отменить роль, используется NONE.

Автоматическая установка роли SET ROLE неявно выполняется при подключении пользователя, если этому пользователю назначена роль по умолчанию.

Функция CURRENT\_ROLE возвращает текущую установленную роль для сеанса, если таковая имеется.

Можно установить только роли, предоставленные непосредственно пользователю, роли, предоставленные другим ролям, нельзя. Вместо этого привилегии, предоставленные роли, которая, в свою очередь, предоставляется другой роли (получателю), будут немедленно доступны любому пользователю, который устанавливает эту вторую роль получателя.

Синтаксис назначения роли по умолчанию:

```
> SET DEFAULT ROLE { role | NONE } [ FOR user@host ]
```

Оператор SET DEFAULT ROLE устанавливает роль по умолчанию для указанного (или текущего) пользователя. Роль по умолчанию автоматически включается при подключении пользователя (неявный оператор SET ROLE выполняется сразу после установления соединения).

Чтобы иметь возможность установить роль по умолчанию, эта роль уже должна быть предоставлена этому пользователю, и необходимы привилегии для включения этой роли (если нет полномочий выполнить SET ROLE X, то нет и полномочий выполнить SET DEFAULT ROLE X). Чтобы установить роль по умолчанию для другого пользователя, необходимо иметь доступ на запись к mysql базе данных.

Чтобы удалить роль пользователя по умолчанию, используйте SET DEFAULT ROLE NONE [ FOR user@host ]. Запись роли по умолчанию не удаляется при удалении или отзыве роли, поэтому, если роль впоследствии будет создана заново или предоставлена, она снова станет ролью пользователя по умолчанию.

Роль по умолчанию хранится в столбце таблицы default\_role представления mysql.user, а также в таблице информационной схемы APPLICABLE\_ROLES, поэтому их можно просмотреть, чтобы узнать, какая роль назначена пользователю по умолчанию.

Информацию о ролях и о том, кому они были предоставлены, можно найти в таблице APPLICABLE\_ROLES информационной схемы, а также в таблице mysql.ROLES\_MAPPING.

Таблица информационной схемы ENABLED\_ROLES показывает включенные роли для текущего сеанса.

## Удаление роли

Синтаксис удаления роли:

```
> DROP ROLE [IF EXISTS] role_name [,role_name ...]
```

Оператор DROP ROLE удаляет одну или несколько ролей. Чтобы использовать этот оператор, у пользователя должна быть глобальная привилегия CREATE USER или привилегия DELETE для базы данных mysql.

DROP ROLE не отключает роли для соединений, которые выбрали их с помощью SET ROLE. Если роль ранее была установлена как роль по умолчанию, DROP ROLE не удаляет запись

роли по умолчанию из таблицы `mysql.user`. Если роль впоследствии будет воссоздана и предоставлена, она снова станет ролью пользователя по умолчанию. Используйте `SET DEFAULT ROLE NONE`, чтобы явно удалить ее.

## Просмотр прав

Синтаксис просмотра прав:

```
> SHOW GRANTS [FOR user|role]
```

Оператор `SHOW GRANTS` предназначен для демонстрации привилегий, предоставленные конкретному пользователю или роли.

Чтобы перечислить привилегии, предоставленные учетной записи, которая используется для подключения к серверу, можно использовать любой из следующих операторов.

```
> SHOW GRANTS;
> SHOW GRANTS FOR CURRENT_USER;
> SHOW GRANTS FOR CURRENT_USER();
```

`SHOW GRANTS` также может использоваться для просмотра привилегий, предоставленных роли:

```
SHOW GRANTS FOR journalist;

+-----+
| Grants for journalist          |
+-----+
| GRANT USAGE ON *.* TO 'journalist'      |
| GRANT DELETE ON `test`.* TO 'journalist' |
+-----+
```

Оператор `SHOW PRIVILEGES` показывает список системных привилегий, которые поддерживает СУБД «КАТРАПС»:

```
> rows in set (0.000 sec)
```

## Администрирование демона ``mariadb``

Запуск серверной версии СУБД «КАТРАПС» и поддержка его функционирования до момента выключения (*shutdown*) осуществляется при помощи команды запуска (демона) ``mariadb``. Функции администрирования для демона ``mariadb`` выполняет команда ``mariadb-admin``.

Используемая команда ``mariadb-admin`` и общий синтаксис выглядят следующим образом:

```
> mariadb-admin [опции] команда [аргумент-команды] [команда [аргумент-команды]] ...
```

Команда ``mariadb-admin`` использует параметры, представленные в Приложении 1 1 1.

## Действия после сбоев и ошибок эксплуатации

СУБД «КАТРАПС» выдает пользователям сообщения об ошибках в следующем формате:

```
> SELECT * FROM x;
> ERROR 1046 (3D000): No database selected
```

В случае ошибки возвращаются три части информации:

- Числовой код ошибки, в данном случае 1046.
- Значение SQLSTATE, состоящее из пяти символов, в данном случае 3D000. Эти коды являются стандартными для ODBC и ANSI SQL.
- Строка, описывающая ошибку, в данном случае *'No database selected'* (в подавляющем большинстве случаев описания ошибки хватает для понимания ее причины).

Как правило, если ошибка вызвана действиями пользователя, после сообщения об ошибке ему необходимо проверить синтаксис запроса, вызвавшего ошибку, при необходимости воспользоваться функцией подсказки/помощи, а после повторить скорректированный запрос.

Если ошибка вызвана системными сбоями, необходимо выполнить диагностику СУБД «КАТРАПС» для выяснения причины сбоя, используя следующий подход:

- просмотреть и проанализировать журналы работы СУБД «КАТРАПС»;
- проверить состояние службы сервера;
- проверить файл системного журнала на предмет обнаружения ошибок;
- проверить, какая программа/процесс использует весь ресурс процессора или блокирует машину, вызывает нехватку памяти, дискового пространства, файловых дескрипторов или какого-либо другого важного ресурса;
  - если проблема в каком-либо процессе, попытаться его принудительно остановить, а затем запустить (при необходимости);
  - если проблема на стороне сервера, выполнить команды: *mysqladmin -u root ping* или *mysqladmin -u root processlist*, чтобы получить от него ответ;
  - если при подключении проблема не связана с сервером, необходимо проверить, нормально ли работает клиент.

## Приложение 1. Параметры команды mariadb-admin

Параметры	Описание
--character-sets-dir=name	Каталог, в котором расположены файлы набора символов.
-C,--compress	Сжатие всей информации, передаваемой между клиентом и сервером, если оба поддерживают сжатие (по умолчанию false).
--connect_timeout=val	Максимальное время в секундах до таймаута соединения. Значение по умолчанию — 43200 (12 часов).
-c val,--count=val	Количество итераций, которые необходимо сделать. Это работает только с -i( --sleep).
--debug[=debug_options],-# [debug_options]	Запись журнала отладки. Типичная строка debug_options — d:t:o,file_name. Значение по умолчанию d:t:o,/tmp/mysqladmin.trace:
--debug-check	Проверка памяти и использования открытого файла при выходе (по умолчанию false).
--debug-info	Вывод отладочной информации, а также статистики использования памяти и ЦП при выходе из программы (по умолчанию false).
--default-auth=plugin	Используемый плагин аутентификации на стороне клиента по умолчанию.
--default-character-set=name	Установка набора символов по умолчанию.
-f,--force	Отмена запроса подтверждения при удалении базы данных (по умолчанию false).
-,--help	Отображение справки.
-h name,--host=name	Имя хоста для подключения.
-l,--local	Чтение файлов данных с клиентского хоста.
-b,--no-beep	Отключить звуковой сигнал при ошибке (по умолчанию false).
-p[password],--password[=password]	Пароль, который будет использоваться при подключении к серверу. Если пароль не указан, он запрашивается с терминала.
--pipe,-W	Подключение к серверу через именованный канал.
-P portnum,--port=portnum	Номер порта, который будет использоваться для подключения, или 0 по умолчанию, в порядке предпочтения: my.cnf, \$MYSQL_TCP_PORT, /etc/services, встроенный стандарт по умолчанию (3306).
--protocol=name	Протокол, используемый для соединения (tcp, socket, pipe, memory).
-r,--relative	Разница между текущим и предыдущим значениями при использовании с ключом -I (по умолчанию false).
-O value,--set-variable=vaue	Изменение значения переменной.
--shutdown_timeout=val	Максимальное количество секунд ожидания отключения сервера. Значение по умолчанию — 3600 (1 час).
-s,--silent	Тихий выход, если не удастся подключиться к серверу.
-i delay,--sleep=delay	Количество секунд задержки между выполнением итераций. Опция --count определяет количество итераций. Если параметр --count не указан, mariadb-admin выполняет команды бесконечно, пока не будет прерван.
-S name,--socket=name	Имя используемого именованного канала для подключений к localhost при использовании socket.
--ssl	Включает TLS (по умолчанию false).
--ssl-ca=name	Определяет абсолютный путь к файлу PEM, который должен содержать один или несколько сертификатов X509 для доверенных центров сертификации (CA), которые будут использоваться для TLS.
--ssl-capath=name	Определяет абсолютный путь к каталогу, содержащему один или несколько файлов PEM, каждый из которых должен содержать один сертификат X509, который доверенный центр сертификации (CA) будет использовать для TLS.
--ssl-cert=name	Определяет абсолютный путь к файлу сертификата X509, который будет использоваться для TLS .
--ssl-cipher=name	Список разрешенных шифров или наборов шифров для использования в TLS.
--ssl-crl=name	Определяет абсолютный путь к файлу PEM, который должен содержать один или несколько отозванных сертификатов X509 для использования в TLS .

--ssl-crlpath=name	Определяет абсолютный путь к каталогу, содержащему один или несколько файлов PEM, каждый из которых должен содержать один отозванный сертификат X509 для использования для TLS.
--ssl-key=name	Определяет абсолютный путь к файлу закрытого ключа, который будет использоваться для TLS.
--ssl-verify-server-cert	Включает проверку сертификата сервера. Эта опция отключена по умолчанию (по умолчанию false).
--tls-version=name	Эта опция принимает список версий протокола TLS, разделенных запятыми. Версия протокола TLS будет включена только в том случае, если она присутствует в этом списке. Все другие версии протокола TLS не допускаются.
-u,--user=name	Задается пользователь для входа, если он не текущий пользователь.
-v,--verbose	Задается дополнительная информация (по умолчанию false).
-V,--version	Вывод информации о версии.
-E,--vertical	Вывод результата вертикально (по умолчанию false).
-w[count],--wait[=count]	Количество повторных попыток установки соединения. По умолчанию — один раз.
--wait-for-all-slaves	Включение отправки последнего события binlog на все подключенные реплики перед завершением работы. Эта опция отключена по умолчанию.
Файлы параметров:	
--print-defaults	Печать списка аргументов программы.
--no-defaults	Игнорирование параметров по умолчанию из любого файла параметров.
--defaults-file=#	Чтение только параметры по умолчанию из данного файла #.
--defaults-extra-file=#	Чтение указанного файла после чтения глобальных файлов.
--defaults-group-suffix=#	Чтение групп опций с заданным суффиксом помимо групп опций по умолчанию.
Группы параметров:	
[mysqldadmin]	Параметры, читаемые mysqldadmin
[mariadb-admin]	Параметры, читаемые mariadb-admin.
[client]	Параметры, читаемые всеми клиентскими программами
[client-server]	Параметры, читаемые всеми клиентскими программами MariaDB и сервером
[client-mariadb]	Опции, читаемые всеми клиентскими программами
Команды:	
create databasename	Создание новой базы данных.
debug	Поручение серверу записи отладочной информации в журнал.
drop databasename	Удаление базы данных и всех ее таблиц.
extended-status	Возврат всех переменных состояния и их значений.
flush-all-statistics	Очистка всех таблиц статистики.
flush-all-status	Сброс статуса и статистики.
flush-binary-log	Очистка двоичного журнала.
flush-client-statistics	Сброс статистики клиентов.
flush-engine-log	Очистка журнала ядра.
flush-error-log	Очистка журнала ошибок.
flush-general-log	Очистка общего журнала запросов.
flush-hosts	Очистка всех кэшей хостов.
flush-index-statistics	Сброс статистики индекса.
flush-logs	Очистка всех журналов.
flush-privileges	Перезагрузка таблицы привилегий.
flush-relay-log	Очистка журнала смены.
flush-slow-log	Очистка журнала медленных запросов.
flush-ssl	Очистка SSL-сертификатов.
flush-status	Очистка переменных состояния.
flush-table-statistics	Очистка статистики таблицы.
flush-tables	Очистка всех таблиц.
flush-threads	Очистка кэша потока.

flush-user-resources	Очистка пользовательских ресурсов.
flush-user-statistics	Сброс статистики пользователей.
kill id,id,...	Остановка потоков MySQL.
password new-password	Смена старого пароля на новый. Новый пароль можно передать в командной строке в качестве следующего аргумента, например, mariadb-admin password "new_password" или можно опустить (если за ним не следует другая команда), и в этом случае пользователю будет предложено ввести пароль. Если пароль содержит специальные символы, его необходимо заключить в кавычки.
old-password new-password	Изменить старый пароль на новый, используя старый формат
ping	Проверка работоспособности mysqld. Статус возврата равен 0, если сервер работает (даже в случае ошибки, например отказа в доступе), 1, если нет.
processlist	Показ списка активных потоков на сервере
reload	Перезагрузка таблицы.
refresh	Очистка всех таблиц, закрытие и открытие файлов журналов.
shutdown	Отключение сервера. При подключении к локальному серверу с помощью socket mariadb-admin ждет, пока файл идентификатора процесса сервера не будет удален, чтобы убедиться, что сервер остановился правильно.
status	Выдача короткого сообщения о состоянии с сервера.
start-all-slaves	Запуск всех реплик.
start-slave	Запуск репликации на сервере-реплике.
stop-all-slaves	Остановка всех реплик.
stop-slave	Остановка репликации на сервере реплики.
variables	Печать доступных переменных.
version	Возврат версии, а также информации о статусе сервера.
Опция команды завершения работы:	
--wait-for-all-slaves	Включение опции означает, что сервер уничтожает потоки дампа двоичного журнала только после того, как все клиентские потоки были уничтожены, и завершает работу только после того, как последний двоичный журнал будет отправлен всем подключенным репликам.

**Приложение 2. Системные привилегии СУБД «КАТРАПС»**

<b>Привилегия/право</b>	<b>Объект доступа</b>	<b>Описание привилегии/права</b>
Alter	Таблицы	Изменение таблиц
Alter routine	Функции, Процедуры	Изменение или удаление сохраненных функций/процедур
Create	Базы данных, Таблицы, Индексы	Создание новых баз данных и таблиц
Create routine	Базы данных	Создание функций/процедур
Create temporary tables	Базы данных	Создание временных таблиц
Create view	Таблицы	Создание новых представлений
Create user	Администрирование	Создание новых пользователей
Delete	Таблицы	Удаление существующих строк
Delete history	Таблицы	Удаление исторических строк таблицы версий
Drop	Базы данных, Таблицы	Удаление баз данных, таблиц и представлений
Event	Администрирование	Создание, изменение, удаление и выполнение событий
Execute	Функции, Процедуры	Выполнение сохраненных процедур
File	Файлы на сервере	Чтение и запись файлов на сервере
Grant option	Базы данных, Таблицы, Функции, Процедуры	Предоставление другим пользователям тех привилегий, которыми обладает пользователь
Index	Таблицы	Создание и удаление индексов
Insert	Таблицы	Добавление данных в таблицу
Lock tables	Базы данных	Блокировка таблиц (вместе с правом SELECT)
Process	Администрирование	Просмотр выполняющихся в данный момент запросов
Proxy	Администрирование	Включение использования прокси
References	Базы данных, Таблицы	Создание ссылок на таблицы
Reload	Администрирование	Перезагрузка и обновление таблиц, журналов и привилегий
Binlog admin	Сервер	Очистка двоичных журналов
Binlog monitor	Сервер	Использование операторов SHOW BINLOG STATUS и SHOW BINARY LOG
Binlog replay	Сервер	Использование оператора BINLOG
Replication master admin	Сервер	Мониторинг подключенных устройств
Replication slave admin	Сервер	Запуск/остановка подчиненного устройства и применение событий binlog
Slave monitor	Сервер	Использование операторов SHOW SLAVE

		STATUS и SHOW RELAYLOG EVENTS
Replication slave	Администрирование	Чтение событий двоичного журнала с основного сервера
Select	Таблицы	Получать строки из таблиц
Show databases	Администрирование	Просмотр всех баз данных
Show view	Таблицы	Просмотр представлений
Shutdown	Администрирование	Выключение сервера
Super	Администрирование	Использование потоков KILL, SET GLOBAL, CHANGE MASTER и др.
Trigger	Таблицы	Использование триггеров
Create tablespace	Администрирование	Создание, изменение и удаление табличных пространств
Update	Таблицы	Обновление существующих строк
Set user	Сервер	Создание представлений и хранимых процедур с другим определителем
Federated admin	Сервер	Выполнение операторов CREATE SERVER, ALTER SERVER, DROP SERVER
Connection admin	Сервер	Обход ограничения на количество подключений и прекращений подключений других пользователей
Read_only admin	Сервер	Выполнение операции записи, даже если объект только для чтения
Usage	Администрирование	Нет привилегий – разрешено только подключение