Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Дисциплина «Структуры и базы данных»

**ОТЧЕТ**

к практическому занятию

на тему:

**«Организация защиты данных в БД»**

|  |
| --- |
| Выполнил студент группы 983871  КАФТАНЧИКОВА Анастасия  Олеговна |
|  |
| (подпись студента, дата) |
| Проверил ассистент ПИКС  Демидович Вероника Константиновна |
|  |
| (подпись преподавателя, дата) |

Минск 2022

**Цель:** изучить обеспечение безопасности MySQL от злоумышленников, шифрование данных в MySQL, разграничение прав доступа в MySQL, привилегии поддерживаемые MySQL.

Размышляя о безопасности в MySQL, необходимо рассмотреть широкий круг возможных тем и то, как они влияют на безопасность сервера MySQL и связанных с ним приложений:

1. Общие факторы, влияющие на безопасность. К ним относятся выбор надежных паролей, отсутствие предоставления пользователям ненужных привилегий, обеспечение безопасности приложений путем предотвращения SQL-инъекций и повреждения данных, а также другие (глава 2 документации [«Общие вопросы безопасности»](https://dev.mysql.com/doc/mysql-security-excerpt/5.7/en/general-security-issues.html)).
2. Безопасность самой установки. Файлы данных, файлы журналов и все файлы приложений вашей установки должны быть защищены, чтобы гарантировать, что они не будут доступны для чтения или записи посторонним лицам. Для получения дополнительной информации обратитесь к главе 3 документации [«Настройка и тестирование после установки»](https://dev.mysql.com/doc/mysql-security-excerpt/5.7/en/postinstallation.html).
3. Контроль доступа и безопасность в самой системе баз данных, включая пользователей и базы данных, которым предоставлен доступ к базам данных, представлениям и хранимым программам, используемым в базе данных. Для получения дополнительной информации обратитесь к главе 4 документации [«Контроль доступа и управление учетными записями»](https://dev.mysql.com/doc/mysql-security-excerpt/5.7/en/access-control.html).
4. Функции, предлагаемые подключаемыми модулями безопасности. См. главу 6 документации [«Плагины безопасности»](https://dev.mysql.com/doc/mysql-security-excerpt/5.7/en/security-plugins.html).
5. Сетевая безопасность MySQL и вашей системы. Безопасность связана с правами для отдельных пользователей, но вы также можете ограничить MySQL, чтобы он был доступен только локально на хосте сервера MySQL или для ограниченного набора других хостов.
6. Убедитесь, что у вас есть адекватные и надлежащие резервные копии файлов базы данных, файлов конфигурации и журналов. Также убедитесь, что у вас есть решение для восстановления и проверьте, что вы можете успешно восстановить информацию из ваших резервных копий. См. [Резервное копирование и восстановление](https://dev.mysql.com/doc/refman/5.7/en/backup-and-recovery.html).

**Обеспечение безопасности MySQL от злоумышленников**

Когда вы подключаетесь к серверу MySQL, вы должны использовать пароль. Пароль не передается в виде открытого текста по соединению. Обработка паролей во время последовательности клиентских соединений была обновлена ​​в MySQL 4.1.1, чтобы быть более безопасной. Злоумышленник, который может перехватить трафик между клиентом и сервером, может взломать пароль. (См. Раздел 2.2.4, [«Хеширование паролей в MySQL»](https://dev.mysql.com/doc/mysql-security-excerpt/5.7/en/password-hashing.html), где представлены различные методы обработки паролей.)

Вся остальная информация передается в виде текста и может быть прочитана любым, кто может наблюдать за соединением. Если соединение между клиентом и сервером проходит через ненадежную сеть, вы можете использовать сжатый протокол, чтобы сделать трафик намного более трудным для расшифровки. Вы также можете использовать внутреннюю поддержку SSL в MySQL, чтобы сделать соединение еще более безопасным. См. Главу 5 [«Использование зашифрованных соединений»](https://dev.mysql.com/doc/mysql-security-excerpt/5.7/en/encrypted-connections.html). В качестве альтернативы используйте SSH для получения зашифрованного соединения TCP / IP между сервером MySQL и клиентом MySQL. Вы можете найти SSH-клиента с открытым исходным кодом по адресу <http://www.openssh.org>, а сравнение SSH-клиентов с открытым исходным кодом и коммерческого использования – по адресу <http://en.wikipedia.org/wiki/Comparison_of_SSH_clients>.

Чтобы обезопасить систему MySQL, вы должны строго соблюдать следующие рекомендации:

1. Требовать, чтобы все учетные записи MySQL имели пароль. Клиентская программа не обязательно знает личность того, кто ее запускает. Для клиент-серверных приложений характерно, что пользователь может указать любое имя пользователя для клиентской программы. Например, любой может использовать программу mysql для подключения как другой человек, просто вызвав ее как mysql -u other\_user db\_name, если other\_user не имеет пароля. Если все учетные записи имеют пароль, подключение с использованием учетной записи другого пользователя становится намного сложнее. Методы установки паролей описаны в Разделе 4.10, [«Назначение паролей учетной записи».](https://dev.mysql.com/doc/mysql-security-excerpt/5.7/en/assigning-passwords.html)
2. Убедитесь, что единственной учетной записью пользователя Unix с правами чтения или записи в каталогах базы данных является учетная запись, используемая для запуска mysqld.
3. Никогда не запускайте сервер MySQL от имени пользователя root Unix. Это чрезвычайно опасно, потому что любой пользователь с привилегией FILE может заставить сервер создавать файлы с правами root (например, ~ root / .bashrc). Чтобы предотвратить это, mysqld отказывается запускаться от имени пользователя root, если это не указано явно с использованием параметра --user = root. Вместо этого mysqld может (и должен) запускаться как обычный непривилегированный пользователь. Вы можете создать отдельную учетную запись Unix с именем mysql, чтобы сделать все еще более безопасным. Используйте эту учетную запись только для администрирования MySQL. Чтобы запустить mysqld от имени другого пользователя Unix, добавьте параметр пользователя, который задает имя пользователя в группе [mysqld] файла параметров my.cnf, где вы указываете параметры сервера. Например:

[mysqld]

user=mysql

Это заставляет сервер запускаться как назначенный пользователь, независимо от того, запускаете ли вы его вручную или с помощью mysqld\_safe или mysql.server. Для получения дополнительной информации см. Раздел 2.5, [«Как запустить MySQL как обычный пользователь»](https://dev.mysql.com/doc/mysql-security-excerpt/5.7/en/changing-mysql-user.html). Запуск mysqld от имени пользователя Unix, отличного от root, не означает, что вам нужно изменить имя пользователя root в таблице user. Имена пользователей для учетных записей MySQL не имеют ничего общего с именами пользователей для учетных записей Unix.

1. Не предоставляйте привилегию FILE пользователям, не являющимся администратором. Любой пользователь, имеющий эту привилегию, может записать файл в любом месте файловой системы с привилегиями демона mysqld. Это включает в себя каталог данных сервера, содержащий файлы, которые реализуют таблицы привилегий. Чтобы сделать операции с привилегиями FILE более безопасными, файлы, созданные с помощью SELECT ... INTO OUTFILE, не перезаписывают существующие файлы и доступны для записи всем.
2. Привилегия FILE может также использоваться для чтения любого файла, который доступен для чтения или доступен пользователю Unix, от имени которого работает сервер. С этой привилегией вы можете прочитать любой файл в таблицу базы данных. Это может быть использовано, например, с помощью LOAD DATA для загрузки / etc / passwd в таблицу, которая затем может быть отображена с помощью SELECT. Чтобы ограничить местоположение, в котором файлы могут быть прочитаны и записаны, установите для системы secure\_file\_priv определенный каталог. См. [Системные переменные сервера.](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html)
3. Не предоставляйте привилегии PROCESS или SUPER пользователям, не являющимся администратором. Вывод mysqladmin processlist и SHOW PROCESSLIST показывает текст любых операторов, выполняемых в настоящее время, поэтому любой пользователь, которому разрешено видеть список процессов сервера, может видеть операторы, запущенные другими пользователями, такими как UPDATE пользователь SET password = PASSWORD (' не безопасно'). mysqld резервирует дополнительное соединение для пользователей с привилегией SUPER, чтобы пользователь root в MySQL мог войти в систему и проверить активность сервера, даже если используются все обычные соединения. Привилегию SUPER можно использовать для разрыва клиентских подключений, изменения работы сервера путем изменения значения системных переменных и управления серверами репликации.
4. Не разрешайте использование символических ссылок на таблицы. (Эту возможность можно отключить с помощью опции --skip-symbolic-links.) Это особенно важно, если вы запускаете mysqld от имени пользователя root, поскольку любой, кто имеет доступ на запись в каталог данных сервера, может удалить любой файл в системе. См. [Использование символических ссылок для таблиц MyISAM в Unix](https://dev.mysql.com/doc/refman/5.7/en/symbolic-links-to-tables.html).
5. Хранимые программы и представления должны быть написаны с использованием рекомендаций по безопасности, обсуждаемых в разделе [Управление доступом к хранимым объектам](https://dev.mysql.com/doc/refman/5.7/en/stored-objects-security.html).
6. Если вы не доверяете своему DNS, вам следует использовать IP-адреса, а не имена хостов в таблицах привилегий. В любом случае вы должны быть очень осторожны при создании записей таблицы привилегий с использованием значений имени хоста, которые содержат символы подстановки.
7. Если вы хотите ограничить количество подключений, разрешенных для одной учетной записи, вы можете сделать это, установив переменную max\_user\_connections в mysqld. Операторы CREATE USER и ALTER USER также поддерживают параметры управления ресурсами для ограничения разрешенного для учетной записи уровня использования сервера. См. [CREATE USER Statement](https://dev.mysql.com/doc/refman/5.7/en/create-user.html) и [ALTER USER Statement.](https://dev.mysql.com/doc/refman/5.7/en/alter-user.html)
8. Если каталог плагинов доступен для записи на сервере, пользователь может записать исполняемый код в файл в каталога, используя SELECT ... INTO DUMPFILE. Этого можно избежать, сделав plugin\_dir доступным только для чтения на сервере, или установив secure\_file\_priv в каталог, в который можно безопасно выполнять записи SELECT.

**Шифрование данных в MySQL**

Шифрование – преобразование читаемого текста в нечитаемый при помощи некоторого алгоритма; применяется для защиты уязвимых данных.

Процесс дешифрования восстанавливает данные в исходное состояние.

В таблице ниже представлены встроенные функции шифрования MySQL.

|  |  |
| --- | --- |
| [AES\_DECRYPT()](https://dev.mysql.com/doc/refman/8.0/en/encryption-functions.html#function_aes-decrypt) | Дешифрация/шифрация при помощи функции AES |
| [AES\_ENCRYPT()](https://dev.mysql.com/doc/refman/8.0/en/encryption-functions.html#function_aes-encrypt) |
| [ASYMMETRIC\_DECRYPT()](https://dev.mysql.com/doc/refman/8.0/en/enterprise-encryption-functions.html#function_asymmetric-decrypt) | Расшифровать зашифрованный текст с помощью закрытого или открытого ключа |
| [ASYMMETRIC\_DERIVE()](https://dev.mysql.com/doc/refman/8.0/en/enterprise-encryption-functions.html#function_asymmetric-derive) | Получить симметричный ключ из асимметричных ключей |
| [ASYMMETRIC\_ENCRYPT()](https://dev.mysql.com/doc/refman/8.0/en/enterprise-encryption-functions.html#function_asymmetric-encrypt) | Зашифровать открытый текст с помощью закрытого или открытого ключа |
| [ASYMMETRIC\_SIGN()](https://dev.mysql.com/doc/refman/8.0/en/enterprise-encryption-functions.html#function_asymmetric-sign) | Создать подпись из дайджеста |
| [ASYMMETRIC\_VERIFY()](https://dev.mysql.com/doc/refman/8.0/en/enterprise-encryption-functions.html#function_asymmetric-verify) | Проверить, что подпись соответствует дайджесту |
| [CREATE\_ASYMMETRIC\_PRIV\_KEY()](https://dev.mysql.com/doc/refman/8.0/en/enterprise-encryption-functions.html#function_create-asymmetric-priv-key) | Создать закрытый ключ |
| [CREATE\_ASYMMETRIC\_PUB\_KEY()](https://dev.mysql.com/doc/refman/8.0/en/enterprise-encryption-functions.html#function_create-asymmetric-pub-key) | Создать открытый ключ |
| [CREATE\_DH\_PARAMETERS()](https://dev.mysql.com/doc/refman/8.0/en/enterprise-encryption-functions.html#function_create-dh-parameters) | Генерация общего секрета DH |
| [CREATE\_DIGEST()](https://dev.mysql.com/doc/refman/8.0/en/enterprise-encryption-functions.html#function_create-digest) | Генерация дайджеста из строки |
| [MD5()](https://dev.mysql.com/doc/refman/8.0/en/encryption-functions.html#function_md5) | Расчёт контрольной суммы MD5 |
| [RANDOM\_BYTES()](https://dev.mysql.com/doc/refman/8.0/en/encryption-functions.html#function_random-bytes) | Вернуть случайный вектор байт |
| [SHA1(), SHA()](https://dev.mysql.com/doc/refman/8.0/en/encryption-functions.html#function_sha1) | Расчёт 160-битной контрольной суммы SHA |
| [SHA2()](https://dev.mysql.com/doc/refman/8.0/en/encryption-functions.html#function_sha2) | Расчёт контрольной суммы SHA2 |
| [VALIDATE\_PASSWORD\_STRENGTH()](https://dev.mysql.com/doc/refman/8.0/en/encryption-functions.html#function_validate-password-strength) | Determine strength of password |

Также необходимо изучить раздел [12.13 Функции шифрования и сжатия](https://dev.mysql.com/doc/refman/8.0/en/encryption-functions.html) официальной документации MySQL.

**Разграничение прав доступа в MySQL**

Права доступа определяют возможность действия над объектом.

Владелец объекта – пользователь, создавший объект.

Остальные пользователи к разным объектам могут иметь разные уровни доступа.

Разрешение на доступ к конкретным объектам БД сохраняется в файле рабочей группы. Файл раб. группы содержит данные о пользователях группы: имена учетных записей пользователей, пароли пользователей, имена групп, в которых входят пользователи.

Полный синтаксис оператора GRANT:

GRANT

priv\_type [(column\_list)]

[, priv\_type [(column\_list)]] ...

ON [object\_type] priv\_level

TO user\_or\_role [, user\_or\_role] ...

[WITH GRANT OPTION]

[AS user

[WITH ROLE

DEFAULT

| NONE

| ALL

| ALL EXCEPT role [, role ] ...

| role [, role ] ...

]

]

}

GRANT PROXY ON user\_or\_role

TO user\_or\_role [, user\_or\_role] ...

[WITH GRANT OPTION]

GRANT role [, role] ...

TO user\_or\_role [, user\_or\_role] ...

[WITH ADMIN OPTION]

object\_type: {

TABLE

| FUNCTION

| PROCEDURE

}

priv\_level: {

\*

| \*.\*

| db\_name.\*

| db\_name.tbl\_name

| tbl\_name

| db\_name.routine\_name

}

Оператор GRANT позволяет системным администраторам назначать привилегии и роли (роль MySQL – это именованная коллекция привилегий; как и в случае с учетными записями пользователей, привилегии ролей могут предоставляться и отзываться), которые могут быть предоставлены учетным записям и ролям пользователей. При определении привилегий и ролей применяются следующие синтаксические ограничения:

1. GRANT не может смешивать предоставление прав и ролей в одном и том же запросе; данный оператор должен предоставлять либо привилегии, либо роли.
2. предложение ON определяет, предоставляет оператор привилегии или роли:
3. с ON, запрос предоставляет привилегии;
4. без ON запрос предоставляет роли.
5. разрешено присваивать учетной записи как привилегии, так и роли, но должны использоваться отдельные операторы GRANT, каждый с синтаксисом, соответствующим тому, что должно быть предоставлено.

Чтобы использовать GRANT, у пользователя должна быть привилегия GRANT OPTION, и привилегии, которые он предоставляет. Когда системная переменная read\_only включена, GRANT дополнительно требует привилегию CONNECTION\_ADMIN или SUPER.

GRANT либо успешно выполняется для всех именованных пользователей и ролей, либо откатывается назад и не оказывает никакого влияния в случае возникновения ошибки. Оператор записывается в двоичный журнал, только если он выполняется для всех именованных пользователей и ролей.

Оператор REVOKE связан с GRANT и позволяет администраторам удалять привилегии учетной записи:

REVOKE

priv\_type [(column\_list)]

[, priv\_type [(column\_list)]] ...

ON [object\_type] priv\_level

FROM user\_or\_role [, user\_or\_role] ...

REVOKE ALL [PRIVILEGES], GRANT OPTION

FROM user\_or\_role [, user\_or\_role] ...

REVOKE PROXY ON user\_or\_role

FROM user\_or\_role [, user\_or\_role] ...

REVOKE role [, role ] ...

FROM user\_or\_role [, user\_or\_role ] ...

Каждое имя учетной записи использует формат, описанный в [Разделе 6.2.4, «Указание имен учетной записи».](https://dev.mysql.com/doc/refman/8.0/en/account-names.html) Каждое имя роли использует формат, описанный в [Разделе 6.2.5, «Указание имен ролей»](https://dev.mysql.com/doc/refman/8.0/en/role-names.html).

Обычно администратор базы данных сначала использует CREATE USER для создания учетной записи и определения ее непривилегированных характеристик, таких как пароль, использует ли он защищенные соединения и ограничения доступа к ресурсам сервера, а затем использует GRANT для определения своих привилегий. ALTER USER может использоваться для изменения непривилегированных характеристик существующих учетных записей.

GRANT поддерживает имена хостов длиной до 255 символов (до 60 символов до версии MySQL 8.0.17). Имена пользователей могут содержать до 32 символов. Имена баз данных, таблиц, столбцов и подпрограмм могут содержать до 64 символов.

**Привилегии, поддерживаемые MySQL**

В следующей таблице приведены допустимые статические типы привилегий priv\_type, которые можно указать для операторов GRANT и REVOKE, а также уровни, на которых может быть предоставлена ​​каждая привилегия. Динамические типы привилегий рассматриваться не будут; информацию о них можно найти в официальной документации по следующей ссылке: <https://dev.mysql.com/doc/refman/8.0/en/grant.html#grant-privileges>.

|  |  |
| --- | --- |
| **Привилегии** | **Значение и допустимые уровни** |
| [ALL [PRIVILEGES]](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_all) | Предоставьте все привилегии на указанном уровне доступа, кроме GRANT OPTION и PROXY |
| [ALTER](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_alter) | Включить использование ALTER TABLE. Уровни: глобальный, база данных, таблица. |
| [ALTER ROUTINE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_alter-routine) | Включить хранимые процедуры, которые будут изменены или удалены. Уровни: глобальный, база данных, рутина. |
| [CREATE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create) | Включить создание базы данных и таблицы. Уровни: глобальный, база данных, таблица. |
| [CREATE ROLE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-role) | Включить создание роли. Уровень: глобальный. |
| [CREATE ROUTINE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-routine) | Включить сохранение хранимой процедуры Уровни: глобальный, база данных. |
| [CREATE TABLESPACE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-tablespace) | Разрешить создание, изменение или удаление табличных пространств и групп файлов журнала. Уровень: глобальный. |
| [CREATE TEMPORARY TABLES](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-temporary-tables) | Включить использование CREATE TEMPORARY TABLE. Уровни: глобальный, база данных |
| [CREATE USER](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-user) | Включить использование CREATE USER, DROP USER, RENAME USER и REVOKE ALL PRIVILEGES. Уровень: глобальный. |
| [CREATE VIEW](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_create-view) | Позвольте представлениям быть созданными или измененными. Уровни: глобальный, база данных, таблица. |
| [DELETE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_delete) | Разрешить использование DELETE. Уровень: глобальный, база данных, таблица. |
| [DROP](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_drop) | Разрешить удаление баз данных, таблиц и представлений. Уровни: глобальный, база данных, таблица. |
| [DROP ROLE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_drop-role) | Разрешить удаление ролей Уровень: глобальный. |
| [EVENT](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_event) | Включить использование событий для планировщика событий. Уровни: глобальный, база данных. |
| [EXECUTE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_execute) | Разрешить пользователю выполнять сохраненные подпрограммы. Уровни: глобальный, база данных, рутина. |
| [FILE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_file) | Позвольте пользователю заставить сервер читать или записывать файлы. Уровень: глобальный. |
| [GRANT OPTION](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_grant-option) | Включите привилегии, которые будут предоставлены или удалены из других учетных записей. Уровни: глобальный, база данных, таблица, рутина, прокси. |
| [INDEX](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_index) | Разрешить создание или удаление индексов. Уровни: глобальный, база данных, таблица. |
| [INSERT](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_insert) | Включить использование INSERT. Уровни: Глобальный, база данных, таблица, столбец. |
| [LOCK TABLES](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_lock-tables) | Включите использование LOCK TABLES для таблиц, для которых у вас есть привилегия SELECT. Уровни: глобальный, база данных. |
| [PROCESS](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_process) | Разрешить пользователю видеть все процессы с помощью SHOW PROCESSLIST. Уровень: глобальный. |
| [PROXY](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_proxy) | Включить прокси пользователя. Уровень: от пользователя к пользователю. |
| [REFERENCES](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_references) | Включить создание внешнего ключа. Уровни: Глобальный, база данных, таблица, столбец. |
| [RELOAD](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_reload) | Разрешить использование операций FLUSH. Уровень: глобальный. |
| [REPLICATION CLIENT](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_replication-client) | Позвольте пользователю спросить, где находятся главный или подчиненный серверы. Уровень: глобальный. |
| [REPLICATION SLAVE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_replication-slave) | Разрешить ведомым репликации для чтения событий двоичного журнала от мастера. Уровень: глобальный. |
| [SELECT](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_select) | Включить использование SELECT. Уровни: Глобальный, база данных, таблица, столбец. |
| [SHOW DATABASES](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_show-databases) | Включите SHOW DATABASES, чтобы показать все базы данных. Уровень: глобальный. |
| [SHOW VIEW](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_show-view) | Включите использование SHOW CREATE VIEW. Уровни: глобальный, база данных, таблица. |
| [SHUTDOWN](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_shutdown) | Разрешить использование mysqladmin shutdown. Уровень: глобальный. |
| [SUPER](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_super) | Включите использование других административных операций, таких как CHANGE MASTER TO, KILL, PURGE BINARY LOGS, SET GLOBAL и mysqladmin debug. Уровень: глобальный. |
| [TRIGGER](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_trigger) | Включить триггерные операции. Уровни: глобальный, база данных, таблица. |
| [UPDATE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_update) | Включить использование UPDATE. Уровни: Глобальный, база данных, таблица, столбец. |
| [USAGE](https://dev.mysql.com/doc/refman/8.0/en/privileges-provided.html#priv_usage) | Синоним «нет привилегий» |

Типы прав доступа:

* **глобальные** привилегии являются административными или применяются ко всем базам данных на данном сервере. Чтобы назначить глобальные привилегии, используйте синтаксис ON \*. \*:

GRANT ALL ON \*.\* TO 'someuser'@'somehost';

GRANT SELECT, INSERT ON \*.\* TO 'someuser'@'somehost';

* **привилегии базы данных** применяются ко всем объектам в данной базе данных. Чтобы назначить привилегии уровня базы данных, используйте ON db\_name.\* . Если вы используете синтаксис ON \* (вместо ON \*.\*), привилегии назначаются на уровне базы данных для базы данных по умолчанию. Ошибка возникает, если нет базы данных по умолчанию. Синтаксис назначения привилегий на БД:

GRANT ALL ON mydb.\* TO 'someuser'@'somehost';

GRANT SELECT, INSERT ON mydb.\* TO 'someuser'@'somehost';

* **привилегии таблицы** применяются ко всем столбцам в данной таблице. Чтобы назначить привилегии на уровне таблицы, используйте синтаксис ON db\_name.tbl\_name:

GRANT ALL ON mydb.mytbl TO 'someuser'@'somehost';

GRANT SELECT, INSERT ON mydb.mytbl TO 'someuser'@'somehost';

Если вы укажете tbl\_name, а не db\_name.tbl\_name, оператор будет применяться к tbl\_name в базе данных по умолчанию. Ошибка возникает, если нет базы данных по умолчанию.

* **права доступа к столбцам** **таблицы** применяются к отдельным столбцам в данной таблице. За каждой привилегией, предоставляемой на уровне столбца, должен следовать столбец или столбцы, заключенные в скобки:

GRANT SELECT (col1), INSERT (col1, col2) ON mydb.mytbl TO 'someuser'@'somehost';

* **привилегии хранимых подпрограмм**; операторы ALTER ROUTINE, CREATE ROUTINE, EXECUTE и GRANT OPTION применяются к хранимым процедурам и функциям. Они могут быть определены на глобальном уровне и уровне базы данных. За исключением CREATE ROUTINE, эти привилегии могут предоставляться на уровне процедуры для отдельных процедур:

GRANT CREATE ROUTINE ON mydb.\* TO 'someuser'@'somehost';

GRANT EXECUTE ON PROCEDURE mydb.myproc TO 'someuser'@'somehost';

* **привилегия PROXY** позволяет одному пользователю быть прокси для другого. Прокси-пользователь олицетворяет или принимает личность прокси-пользователя; то есть он принимает привилегии проксируемого пользователя.

GRANT PROXY ON 'localuser'@'localhost' TO 'externaluser'@'somehost';

Когда PROXY предоставлен, это должна быть единственная привилегия, указанная в операторе GRANT, а единственная разрешенная опция для WITH – это опция GRANT OPTION.

**Предоставление привилегий ролям**

Чтобы избежать предоставления привилегий отдельным учетным записям, возможно создание роли в качестве имен для необходимых наборов привилегий. Это позволяет легко предоставлять необходимые привилегии учетным записям пользователей, предоставляя соответствующие роли.

Для создания роли используется оператор CREATE ROLE:

CREATE ROLE 'app\_developer', 'app\_read', 'app\_write';

Имена ролей очень похожи на имена учетных записей пользователей и состоят из пользовательской части и хост-части в формате 'user\_name' @ 'host\_name'. Часть хоста, если не указана, по умолчанию имеет значение «%». Части пользователя и хоста могут не заключаться в кавычки, если они не содержат специальных символов, таких как **-** или **%**. В отличие от имен учетных записей, пользовательская часть имен ролей не может быть пустой.

Чтобы назначить привилегии ролям, необходимо использовать оператора GRANT, используя тот же синтаксис, что и для назначения привилегий учетным записям пользователей:

GRANT ALL ON app\_db.\* TO 'app\_developer';

GRANT SELECT ON app\_db.\* TO 'app\_read';

GRANT INSERT, UPDATE, DELETE ON app\_db.\* TO 'app\_write';

Теперь предположим, что изначально требуется одна учетная запись разработчика, две учетные записи пользователей, которым необходим доступ только для чтения, и одна учетная запись пользователя, для которой требуется доступ для чтения / записи. Используя CREATE USER, создадим учетные записи:

CREATE USER 'dev1'@'localhost' IDENTIFIED BY 'dev1pass';

CREATE USER 'read\_user1'@'localhost' IDENTIFIED BY 'read\_user1pass';

CREATE USER 'read\_user2'@'localhost' IDENTIFIED BY 'read\_user2pass';

CREATE USER 'rw\_user1'@'localhost' IDENTIFIED BY 'rw\_user1pass';

Чтобы назначить каждой учетной записи пользователя свои требуемые привилегии, можно использовать оператор GRANT, представленный выше, но уже с перечислением индивидуальных привилегий для каждого пользователя. Однако, вместо этого можно использовать альтернативный синтаксис GRANT, который разрешает предоставление ролей, а не привилегий:

GRANT 'app\_developer' TO 'dev1'@'localhost';

GRANT 'app\_read' TO 'read\_user1'@'localhost', 'read\_user2'@'localhost';

GRANT 'app\_read', 'app\_write' TO 'rw\_user1'@'localhost';

Оператор GRANT для учетной записи rw\_user1 предоставляет роли для чтения и записи, которые в совокупности обеспечивают необходимые права на чтение и запись.

Синтаксис GRANT для предоставления ролей учетной записи отличается от синтаксиса для предоставления привилегий: есть предложение ON для назначения привилегий, тогда как нет условия ON для назначения ролей. Поскольку синтаксис различен, нельзя смешивать назначение привилегий и ролей в одном и том же запросе. (Разрешено назначать как привилегии, так и роли учетной записи, но нужно использовать отдельные операторы GRANT, каждый с синтаксисом, соответствующим тому, что должно быть предоставлено). Начиная с MySQL 8.0.16, роли не могут предоставляться анонимным пользователям.

Роль при создании заблокирована, не имеет пароля, и ей назначается подключаемый модуль аутентификации по умолчанию (эти атрибуты роли могут быть изменены позже с помощью инструкции ALTER USER пользователями, имеющими глобальную привилегию CREATE USER.)

Чтобы проверить привилегии, назначенные учетной записи, используйте операторы SHOW GRANTS:

SHOW GRANTS FOR 'dev1'@'localhost';

Однако это показывает каждую предоставленную роль без «расширения» ее до привилегий, которые представляет роль. Чтобы также показать привилегии ролей, нужно указать предложение USING, в котором будут указаны роли, для которых нужно отобразить привилегии:

SHOW GRANTS FOR 'dev1'@'localhost' USING 'app\_developer';

**Выводы:** изучил обеспечение безопасности MySQL от злоумышленников, шифрование данных в MySQL, разграничение прав доступа в MySQL, привилегии поддерживаемые MySQL.