

Привилегии в MySQL

Для работы с базами данных в MySQL необходим пользователь, наделённый такими правами. То есть при подключении к базе данных Вы должны указывать логин пользователя и его пароль, и если доступ ему открыт, то он получит определённые права.

В MySQL существуют три группы привилегий: **данные, структура, администрирование**. Первая группа связана с изменением записей в таблицах, вторая группа связана с изменением структуры баз данных, а третья связана с администрированием, как бы это очевидно не звучало.

Теперь перейдём к рассмотрению каждой группы отдельно. И начнём с **прав на управление данными** в таблицах.

- **SELECT** - эта привилегия позволяет делать выборку (вытаскивание) записей из таблиц баз данных.
- **INSERT** - привилегия, которая необходима для добавления новых записей в таблицу.
- **UPDATE** - право, позволяющее обновлять записи в таблице.
- **DELETE** - эта привилегия позволяет удалять записи из таблицы.
- **FILE** - разрешает делать выборку записей и записывать данные в файл, а также считывать их оттуда.

Теперь перейдём к **привилегиям пользователей MySQL**, позволяющие изменять структуру таблицы и базы данных.

- **CREATE** - привилегия, позволяющая создавать новые базы данных, а также новые таблицы в базе данных.
- **ALTER** - привилегия, позволяющая переименовывать таблицы, вставлять новые поля в таблицу, удалять поля из таблицы, а также модифицировать их.
- **INDEX** - разрешает создавать индекс по определённому полю и удалять его. О том, что это такое и для чего нужно мы поговорим в одной из следующих статей.
- **DROP** - право, которое позволяет удалять либо таблицы, либо целые базы данных.
- **CREATE TEMPORARY TABLES** - возможность создавать временные таблицы, которые хранятся во время сессии, а после окончания сессии данная таблица автоматически удаляется.

И последняя группа привилегий - это **привилегии, связанные с администрированием** баз данных.

- **GRANT** - привилегия, которая позволяет создавать новых пользователей, а также менять права у существующих. Тут есть очень важная деталь: нельзя изменять значения привилегий, которыми сам не обладаешь. То есть если человек обладает привилегией **GRANT**, но не обладает привилегией **SELECT**, то он не может новым пользователям дать привилегию **SELECT** и это вполне логично.
- **SUPER** - позволяет использовать команду "**kill**", то есть убить поток. Поток - это текущее подключение другого пользователя к базе данных.
- **PROCESS** - привилегия, позволяющая выполнить команду "**processlist**", которая показывает список потоков.
- **RELOAD** - позволяет открывать и закрывать файлы журналов, а также перечитывать таблицы привилегий пользователей.
- **SHUTDOWN** - привилегия, позволяющая выполнить команду "**shutdown**", отключающая работу сервера.
- **SHOW DATABASES** - разрешает просматривать все существующие базы данных.
- **REFERENCES** - данная привилегия ещё не доступна, а только зарезервирована для использования в будущем.
- **LOCK TABLES** - позволяет блокировать таблицы от указанных потоков.
- **EXECUTE** - позволяет запускать хранимые процедуры.
- **REPLICATION CLIENT** - даёт право получать местонахождение ведущего (**master**) и ведомых (**slaves**) серверов.
- **REPLICATION SLAVE** - это привилегия, позволяющая читать ведомым журнала ведущего сервера.

И, наконец, специальные привилегии, связанные с ограничением на доступные ресурсы:

- **MAX QUERIES PER HOUR** - максимальное количество запросов в час, которое может отправить пользователь.
- **MAX UPDATES PER HOUR** - максимальное количество команд в час, которые каким-либо образом изменяют либо таблицу, либо базу данных.
- **MAX CONNECTIONS PER HOUR** - максимальное количество подключений в час, которое может сделать пользователь.

Если значение вышеназванных пределов равны "**0**", то ресурсы для пользователя не ограничены.

Создание нового пользователя в MySQL

Ранее мы вносили все изменения в настройки MySQL под root-пользователем, имея полный доступ ко всем базам данных. Однако для случаев, когда могут потребоваться более жесткие ограничения, есть способы создания пользователей с особыми наборами прав доступа.

Давайте начнем с создания нового пользователя из консоли MySQL:

```
CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';
```

К сожалению, на данном этапе пользователь "newuser" не имеет прав делать что-либо с базами данных. На самом деле, даже если пользователь "newuser" попытается залогиниться (с паролем "password"), он не попадет в консоль MySQL.

Таким образом, первое, что нам необходимо сделать, это предоставить пользователю доступ к информации, которая ему потребуется.

```
GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';
```

Звездочки в этой команде задают базу и таблицу, соответственно, к которым у пользователя будет доступ. Конкретно эта команда позволяет пользователю читать, редактировать, выполнять любые действия над всеми базами данных и таблицами.

Поле завершения настройки прав доступа новых пользователей, убедитесь, что вы обновили все права доступа:

```
FLUSH PRIVILEGES;
```

Теперь ваши изменения вступят в силу.

Назначение привилегий

Для назначения прав конкретному пользователю можно использовать следующую схему:

```
GRANT [тип прав] ON [название базы данных].[название таблицы] TO '[имя пользователя]@'localhost';
```

Если вы хотите дать доступ к любой базе данных или к любой таблице, поставьте звездочку (*) вместо названия базы данных или таблицы.

Каждый раз, когда вы изменяете права доступа, не забудьте использовать команду Flush Privileges.

Лишения прав доступа практически идентично их назначению:

```
REVOKE [тип прав] ON [название базы данных].[название таблицы] FROM '[имя пользователя]@'localhost';
```

По аналогии с использованием команды DROP для удаления базы данных, вы можете использовать эту команду и для удаления пользователя.

```
DROP USER 'demo'@'localhost';
```

Для тестирования учетной записи созданного пользователя, разлогиньтесь с помощью команды:

```
quit
```

и залогиньтесь снова, введя в термине следующую команду:

```
mysql -u [имя пользователя]-p
```