MySQL / MariaDB

- Справочник по MySQL / MariaDB

?????????? ?? MySQL / MariaDB

Основные команды для администрирования и работы с базами данных MySQL и MariaDB.

????????? ? ?????????

??? 1: ????????? ???????

Для Debian/Ubuntu, MariaDB является заменой MySQL по умолчанию.

```
sudo apt update && sudo apt install -y mariadb-server
```

??? 2: ?????????? ?????????

После установки запустите скрипт безопасной настройки. Он поможет установить пароль для root, удалить анонимных пользователей и тестовую базу данных.

```
sudo mysql_secure_installation
```

??? 3: ??????????? ? ???????

Для подключения к консоли MySQL/MariaDB используйте следующую команду.

```
sudo mysql -u root -p
```

?????????? ?????? ??????

???????? ?? (`CREATE DATABASE`)

Команда для создания новой базы данных. Рекомендуется сразу указывать кодировку `utf8mb4` для полной поддержки Unicode.

```
CREATE DATABASE my_new_app CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

???????? ???? ??

```
SHOW DATABASES;
```

???????? ?? (`DROP DATABASE`)

Внимание! Команда необратима.

```
DROP DATABASE my_new_app;
```

?????????? ??????????????

???????? ???????????

Создает нового пользователя с паролем. `%` означает, что пользователь сможет подключаться с любого хоста. Для большей безопасности можно указать `localhost` или конкретный IP.

```
CREATE USER 'newuser'@'%' IDENTIFIED BY 'a_very_strong_password';
```

????????????? ????

Дает пользователю все права на определенную базу данных.

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

?????????? ?????????

После создания пользователя и выдачи прав необходимо обновить привилегии.

```
FLUSH PRIVILEGES;
```

?????? ? ??????? (SQL)

Основные команды для взаимодействия с таблицами и данными внутри базы. Все команды выполняются в консоли MySQL.

????? ???? ?????? ? ???????? ??????

Сначала выберите базу данных для работы:

```
USE my_new_app;
```

Посмотреть все таблицы в текущей базе:

```
SHOW TABLES;
```

Посмотреть структуру конкретной таблицы:

```
DESCRIBE users;
```

????? ? ????????? ??????

Найти всех пользователей с именем 'Ivan':

```
SELECT * FROM users WHERE name = 'Ivan';
```

Изменить email пользователя с ID 123:

```
UPDATE users SET email = 'new.email@example.com' WHERE id = 123;
```

?????????? ??????????? ??????? ? ????

Позволяет экспортировать результат выполнения запроса `SELECT` в текстовый файл на сервере.

Важно: По умолчанию из соображений безопасности MySQL может ограничивать каталоги для экспорта. Вам может потребоваться настроить переменную `secure_file_priv` в конфигурационном файле `my.cnf`.

```
SELECT id, name, email FROM users WHERE registration_date > '2023-01-01'
INTO OUTFILE '/var/lib/mysql-files/recent_users.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

???????? ? ?????? ??????

???????? ???????

Команда для проверки таблицы или таблиц на наличие ошибок.

```
CHECK TABLE my_table, another_table;
```

?????? ???????

Пытается исправить поврежденную таблицу. **В основном эффективно для старого движка MyISAM.** Для InnoDB лучшей стратегией является восстановление из резервной

копии.

```
REPAIR TABLE my_table;
```

????????? ???????????? ? ??????????????

Эти команды выполняются из обычной командной строки Linux, а не из консоли MySQL.

????????? ??????????? (Backup)

`mysqldump` создает текстовый файл с SQL-командами для воссоздания базы.

```
sudo mysqldump -u root -p my_database_name > backup.sql
```

????????????? (Restore)

Для восстановления сначала нужно создать пустую базу данных, а затем импортировать в нее данные из бэкапа.

1. Создаем пустую базу в консоли MySQL:

```
CREATE DATABASE my_database_name;
```

2. Импортируем данные в командной строке Linux:

```
sudo mysql -u root -p my_database_name < backup.sql
```