**MySql Server**

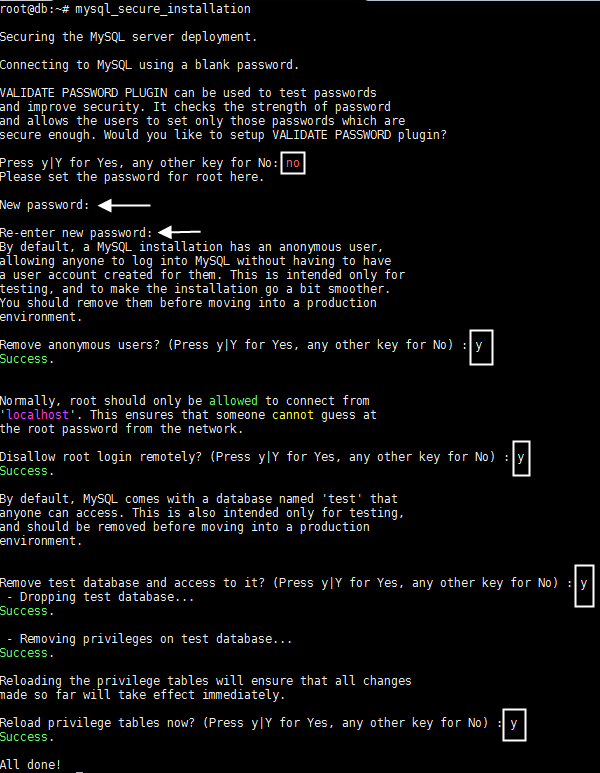
**MySQL is an open-source database management system, commonly installed as part of the popular LAMP (Linux, Apache, MySQL, PHP/Python/Perl) stack. It uses a relational database and SQL (Structured Query Language) to manage its data.**

**update your package index, install the mysql-server package, and then run the included security script.**

sudo apt update

sudo apt install mysql-server

sudo mysql\_secure\_installation



systemctl start mysql

sudo systemctl enable mysql

**At the command prompt, run the following command to launch the mysql shell and enter it as the root user:**

mysql -u root -p

**When you’re prompted for a password, enter the one that you set at installation time, or if you haven’t set one, press Enter to submit no password.**

**The following mysql shell prompt should appear:**

*mysql>*

**MySQL Create Database**

CREATE DATABASE employees;

SHOW DATABASES;

USE employees;

**MySQL Drop Database**

DROP DATABASE employees;

**MySQL CREATE TABLE**

*Syntax:*

*CREATE TABLE table\_name (column\_name column\_type...);*

**Example:**

**Here, we will create a table named "cus\_tbl" in the database "customers".**

CREATE TABLE cus\_tbl(

id INT NOT NULL AUTO\_INCREMENT,

firstname VARCHAR(100) NOT NULL,

surname VARCHAR(100) NOT NULL,

age VARCHAR (100) NOT NULL,

PRIMARY KEY ( id )

);

**Note:**

**Here, NOT NULL is a field attribute and it is used because we don't want this field to be NULL. If you will try to create a record with NULL value, then MySQL will raise an error.**

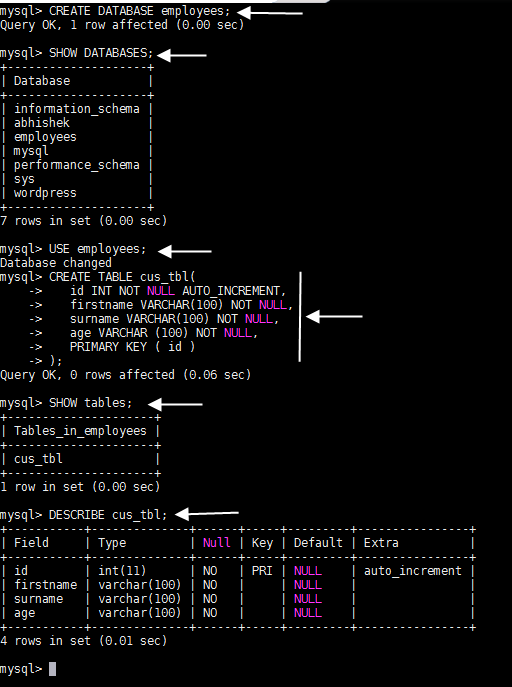
**The field attribute AUTO\_INCREMENT specifies MySQL to go ahead and add the next available number to the id field.PRIMARY KEY is used to define a column as primary key. You can use multiple columns separated by comma to define a primary key.**

**See the created table:**

SHOW tables;

**See the table structure:**

DESCRIBE cus\_tbl;

****

**MySQL ALTER Table**

**The ALTER statement is always used with "ADD", "DROP" and "MODIFY" commands according to the situation.**

1) ADD a column in the table

ALTER TABLE cus\_tbl ADD phone varchar(40) NOT NULL;

2) Add multiple columns in the table

ALTER TABLE cus\_tbl

ADD address varchar(100) NOT NULL

AFTER surname,

ADD salary int(100) NOT NULL

AFTER age;

**In this example, we add two new columns 'address" after surname and "salary" in after age column.**

**See the recently added columns:**

SELECT\* FROM cus\_tbl;

3) MODIFY column in the table

ALTER TABLE cus\_tbl

MODIFY surname varchar(50) NULL;

**In this example, we modify the column surname to be a data type of varchar(50) and force the column to allow NULL values.**

4) DROP column in table

ALTER TABLE cus\_tbl

DROP COLUMN address;

5) RENAME column in table

ALTER TABLE cus\_tbl

CHANGE COLUMN surname title

varchar(20) NOT NULL;

**In this example, we will change the column name "surname" to "title".**

6) RENAME table

ALTER TABLE cus\_tbl

RENAME TO cus\_table;

**MySQL TRUNCATE Table**

**MYSQL TRUNCATE statement removes the complete data without removing its structure.**

TRUNCATE TABLE cus\_tbl;

**See the table:**

SELECT\* FROM cus\_tbl;

**it will show empty table.**

**MySQL DROP Table**

DROP TABLE cus\_tbl;

**MySQL Insert Query**

**MySQL insert query is used to insert records into table. For example:**

**1) Insert Query**

insert into cus\_tbl values(1,'abhi','zarkar','chandrapur','23','200000');

**2) MySQL Update Query**

update cus\_tbl set firstname='abhijit', address='nagpur' where id=1;

**3) MySQL Delete Query**

delete from cus\_tbl where id=1;

**4)inserting multiple records**

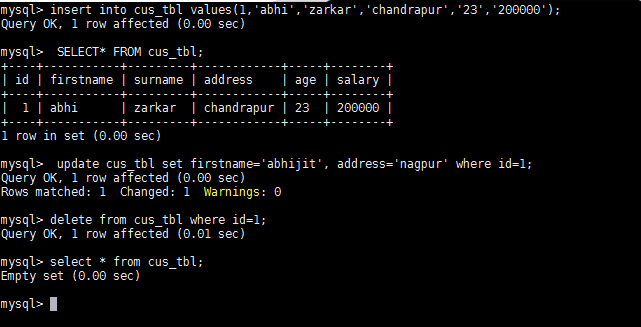
INSERT INTO cus\_tbl

(id, firstname, surname, address, age, salary ) VALUES

(1, 'Abhijit', 'zarkar','chandrapur','23','20000'),

(2, 'devendra', 'ramtekar','nagpur','25','20000'),

(3, 'abhishek', 'dahare','ramtek','24','20000');



**How to Create a New User**

mysql> CREATE USER 'abhijit'@'localhost' IDENTIFIED BY 'temp123';

mysql> GRANT ALL PRIVILEGES ON \* . \* TO 'abhijit'@'localhost';

**(\*.\*--> \*=database, \*=tables)**

**always be sure to reload all the privileges.**

mysql> FLUSH PRIVILEGES;

**To provide a specific user with a permission, you can use this framework**:

GRANT type\_of\_permission ON database\_name.table\_name TO ‘username’@'localhost’;

delete a user

mysql> DROP USER ‘abhijit’@‘localhost’;

**Use below SQL query to see list of mysql users.**

mysql> SELECT User FROM mysql.user;

**If you want to list of MySQL user information, including user permission information and all users data.**

mysql> select User, Host, Password from mysql.user;

**usful commands in MySql**

**How to set MySQL Root password?**

mysqladmin -u root password YOURNEWPASSWORD

**How to Change MySQL Root password?**

mysqladmin -u root -p123456 password 'xyz123'

**How to check all the running Process of MySQL server?**

mysqladmin -u root -p processlist

OR

mysql> show processlist

**How to kill Sleeping MySQL Client Process?**

mysql> show processlist

+----+------+-----------+----+---------+------+-------+------------------+

| Id | User | Host | db | Command | Time | State | Info |

+----+------+-----------+----+---------+------+-------+------------------+

| 5 | root | localhost | | Sleep | 14 | | |

| 8 | root | localhost | | Query | 0 | | show processlist |

+----+------+-----------+----+---------+------+-------+------------------+

mysql> kill 5

**Set up a remote MySQL database connection**

server.mit.com 192.168.72.91

client.mit.com 192.168.72.95

**on server (server.mit.com)**

**Create the remote connection**

**On your database server, as a user with root privileges, open your MySQL configuration file.**

vi /etc/mysql/mysql.conf.d/mysqld.cnf.

**Search the configuration file for bind-address.**

*bind-address = 0.0.0.0*

**Restart the MySQL service:**

service mysql restart

**Use a GRANT command in the following format to enable access for the remote user.**

create user test@192.168.72.95 identified by 'temp';

grat all privileges on \*.\* to test@192.168.72.95;

OR

create user 'test2'@'%' identified by 'temp';

% = for all client servers.

**on client (client.mit.com)**

**first you need to install mysql-client**

apt install mysql-client -you

**Test the connection remotely**

**On client side, enter the following command to verify the connection works:**

mysql -u <local database username> -h <database server ip address> -p

mysql -u test -p -h 192.168.72.91

mysql>

**For phpmyadmin**

 if you want to manage MySQL or MariaDB databases on a remote server via phpMyAdmin web interface,

vi **/etc/phpmyadmin/config.inc.php**

Then change the line that looks like the one below

$cfg['Servers'][$i]['host'] = '$dbserver';

To

$cfg['Servers'][$i]['host'] = '192.168.77.91';

**how to take backup of database and restore database.**

**for example we have on database 'employees'**

**we have to run following command**

mysqldump -u[user name] -p[password] [database name] > [dump file]

mysqldump -u root -p employees > employeess.sql

and for restore the database

**for example we have database 'custumers' and we want store all the 'employees' data in 'custumers'**

**so we have to run following command**

mysql -u root -p custumers < employees.sql