**Projektziele:**

* Erkennen von Tastendrücken und ansteuern einer LED
* Bluetoothverbindung zwischen zwei RaspberryPi und "verbinden" des Tasters mit der jeweils anderen LED
* Abspeichern der Aktionen in einer MySQL Datenbank auf dem Raspberry Pi

## **Basisaufgabe: Erkennen von Tastendrücken und ansteuern einer LED**

Zuerst wird das Erkennen von Tastendrücken und das Ansteuern einer LED geübt. Dazu wurde ein einfaches Python-Programm zur Tasterabfrage geschrieben. Der Status soll dazu ausgelesen werden und sobald der Taster gedrückt wird, soll die LED leuchten. Wird der Taster nicht mehr gedrückt, so soll die LED auch aufhören zu leuchten.

import RPi.GPIO as GPIO #GPIO Library um Pins anzusteuern

import time             #Library für Verzögerungsfunktionen

GPIO.setmode(GPIO.BCM)  #GPIO Pins sollen mit der GPIO Nummerierung angesprochen werden

GPIO.setup(23, GPIO.OUT)#GPIO Pin 23 soll als Output definiert werden

GPIO.setup(24,GPIO.IN)

#Beispiel: Bei Tastendruck LED leuchten lassen

# Endlosschleife

while True:

    if GPIO.input(24) == 0:

        # Ausschalten

        GPIO.output(23, GPIO.LOW)

    else:

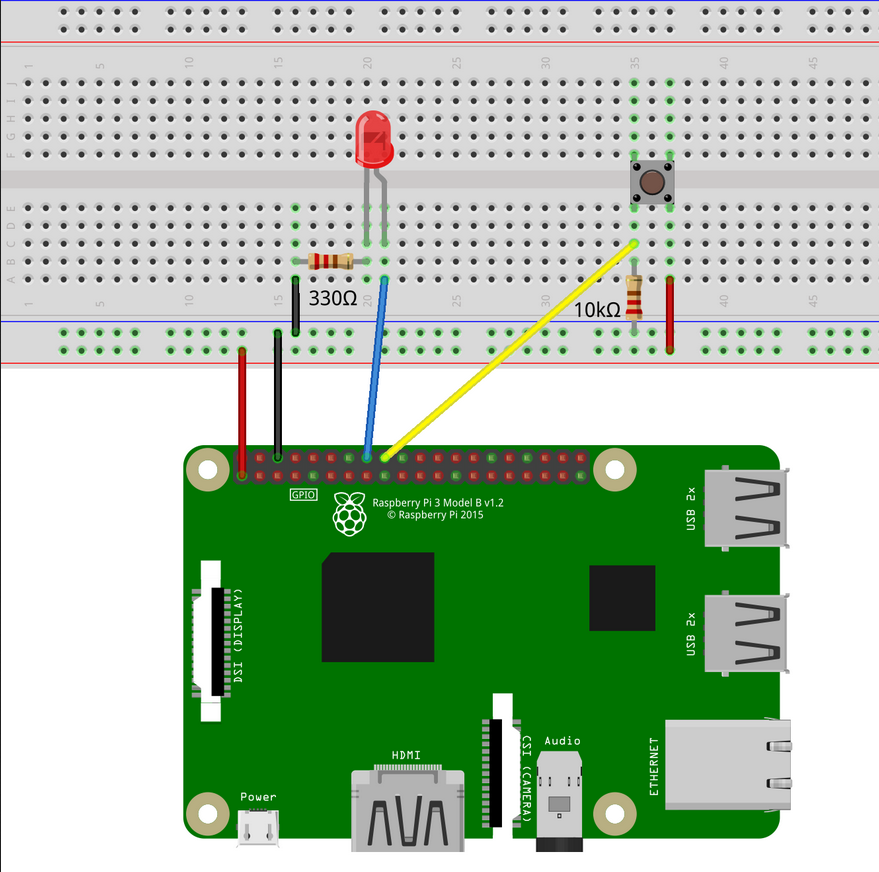
        # Einschalten

        GPIO.output(23, GPIO.HIGH)

time.sleep(0.1)

Pin 23 liefert eine 3.3V Spannung, wenn er auf HIGH gesetzt wird. Mit dieser Spannung wird eine LED zum Leuchten gebracht. Ein Schutzwiderstand sollte in Serie zur LED geschalten werden, damit der Stromfluss begrenzt wird (siehe Versuchsaufbau).

So ist der Versuchsaufbau aufzubauen:



In dieser Abbildung kann man auch sehen, dass die LED mit einem Widerstand in Serie geschalten ist. Um den Strom durch den Taster zu begrenzen, wurde ebenfalls ein Widerstand in Serie geschalten. Die Spannung, welche vom Taster geschaltet wird, kommt vom Pin 1 und beträgt 3.3V. Mehr Spannung würde die GPIO Pins des RPI beschädigen.

## Bluetoothverbindung zwischen zwei Raspberry Pi herstellen und "verbinden" eines Tasters mit der jeweils einer LED

Zum Erstellen der Bluetooth Verbindung wird die Library **PyBluez** verwendet. Dabei wird eine Client-Server Architektur mit zwei Raspberry Pis aufgebaut.

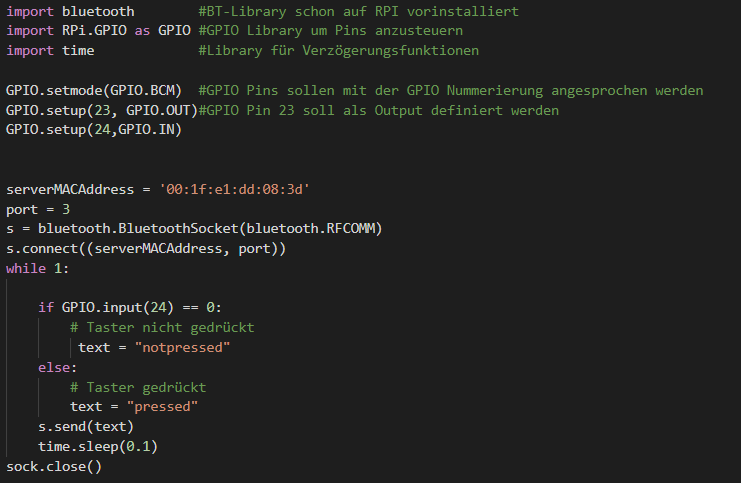
**Vorgehensweise:**

* Ein Raspberry Pi fungiert als Client
* Ein Raspberry Pi fungiert als Server

Das heißt, dass einer auf die Verbindungsanfrage vom anderen wartet. Dazu wurden folgende Befehle in einem Python-Programm angewandt:

1. Der Client

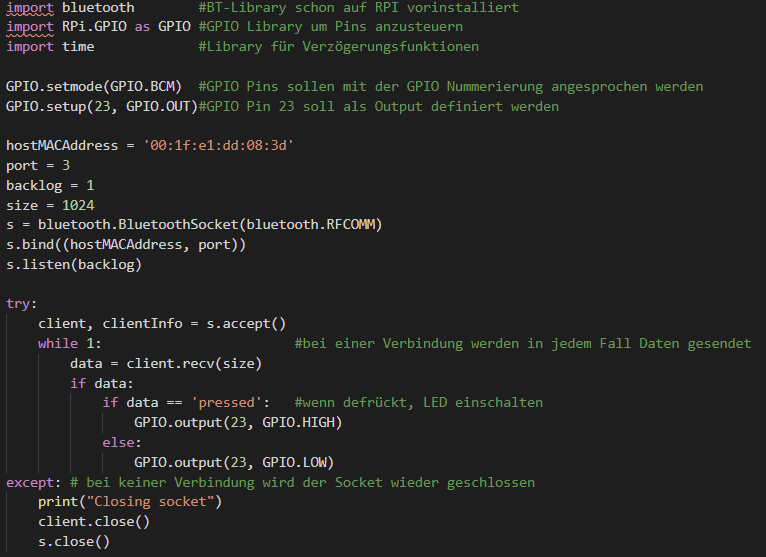
Benötigt die Bluetooth-MAC-Adresse des Servers. Diese wird in einer Variablen abgespeichert. Dann wird ein Bluetooth-Socket geöffnet und versucht, mit der Server-MAC-Adresse eine Verbindung aufzubauen. Danach wird, wie untern im Code gezeigt, abgefragt, ob der Taster gedrückt wurde oder nicht, und dementsprechend ein Text definiert. Der Tastendruck wird dann als Text an den Server gesendet.



**Anmerkung**: für mehrere Taster kann der Code einfach erweitert werden.

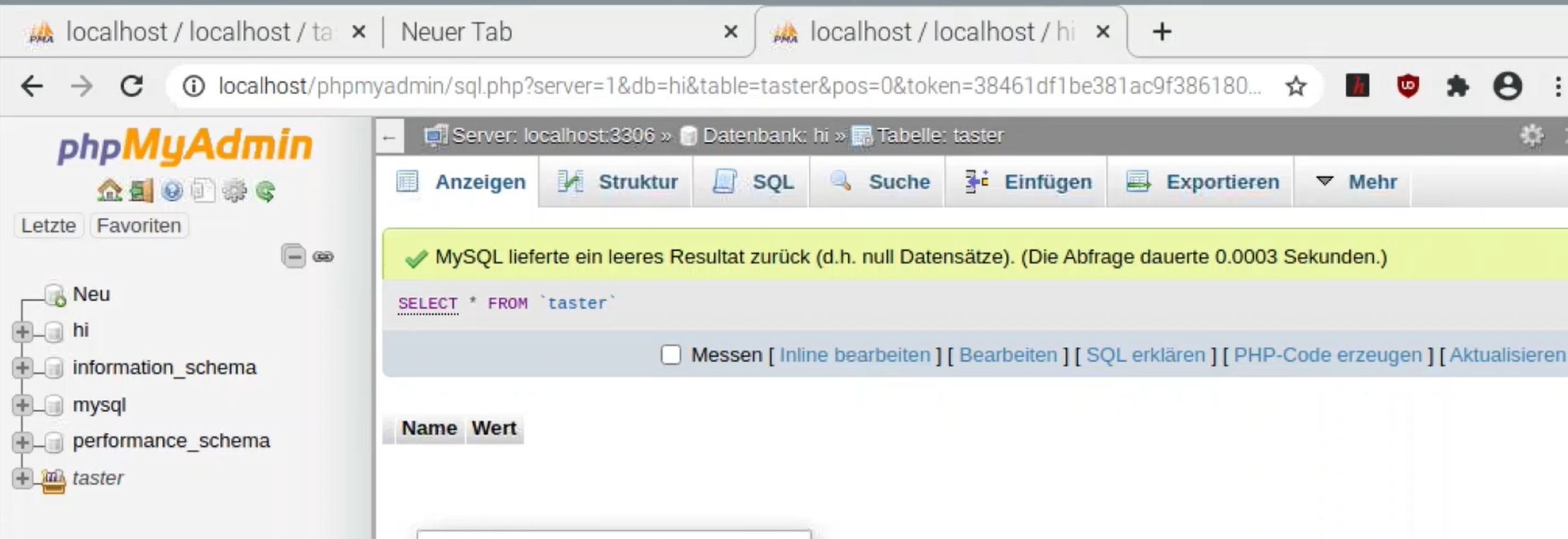
1. Der Server

Der Server benötigt die eigene MAC-Adresse. Nach erfolgreichem Verbindungsaufbau bekommt der Server entweder die Meldung „pressed“ oder „notpressed“ und aktiviert dementsprechend mit **LED ein** oder **LED aus.**

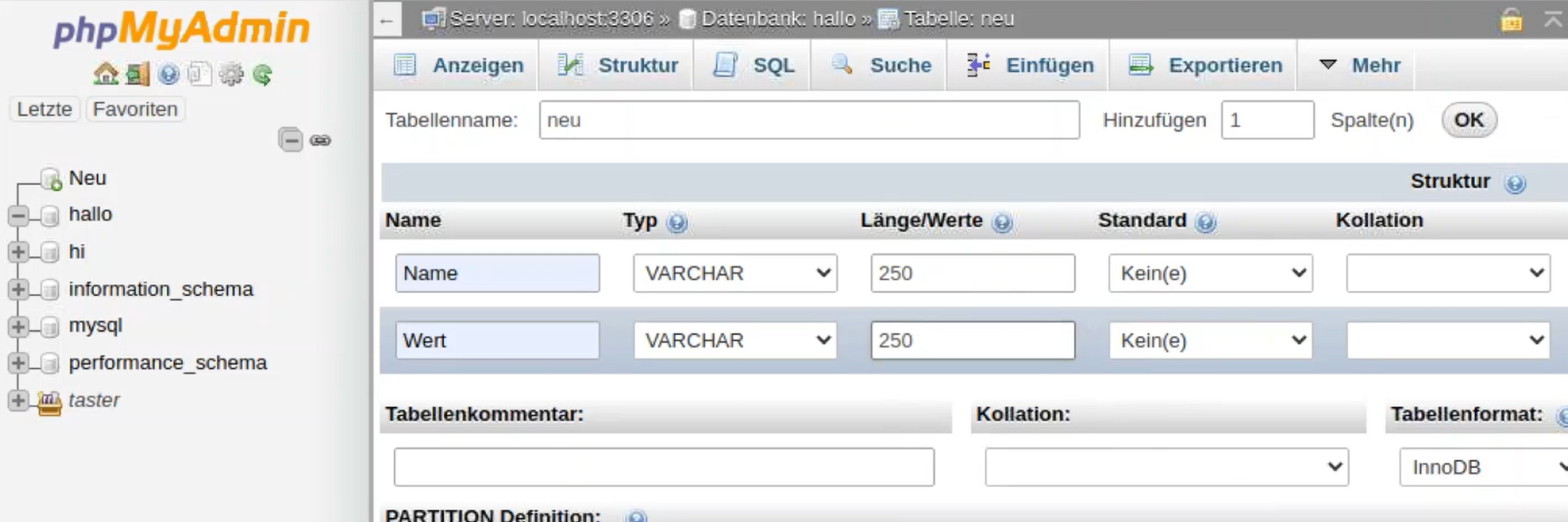


**Wichtig:** Beide Codeteile konnten nicht getestet werden. Daher kann es sein, dass der Code noch angepasst werden muss.

## Abspeichern der Aktionen in einer MySQL Datenbank auf dem Raspberry Pi



* Im ersten Schritt wird eine Datenbank auf der Webseite phpmyadmin erstellt



* 2 Spalten mit den Namen Name und Wert
* Datentyp VARCHAR

Code

Teil 1



Code Teil 2

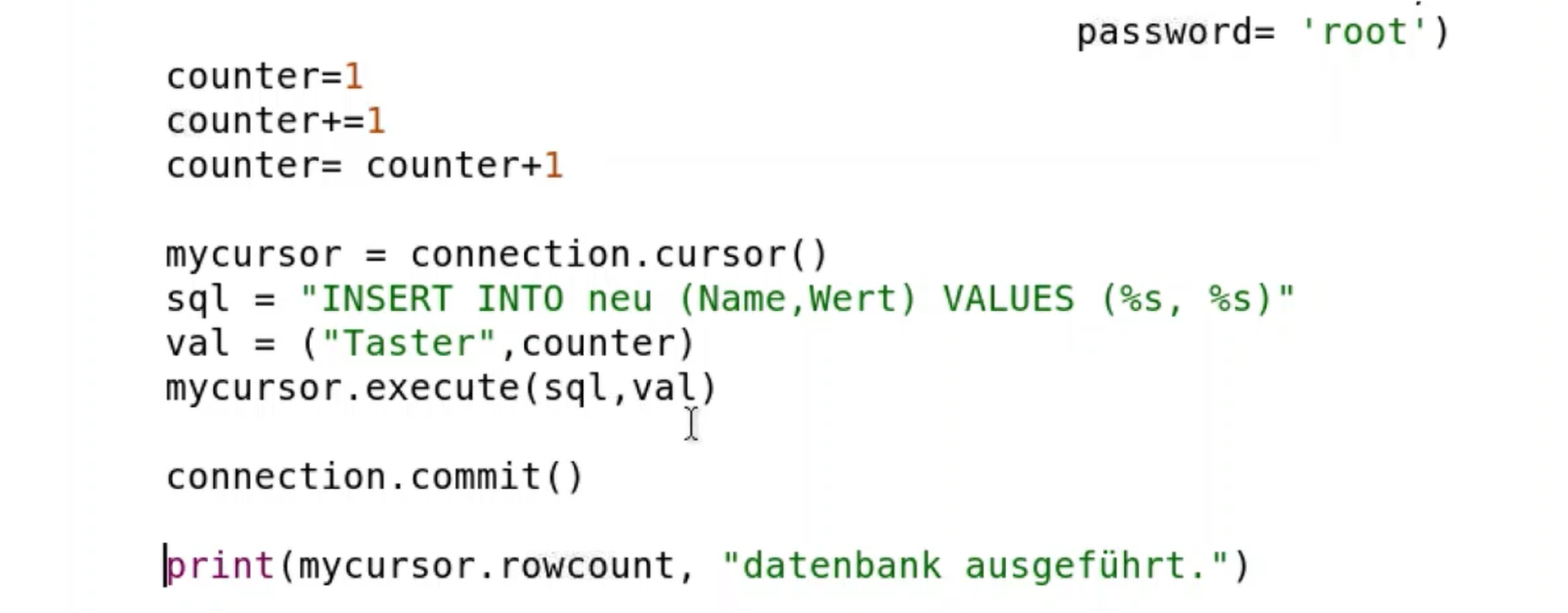
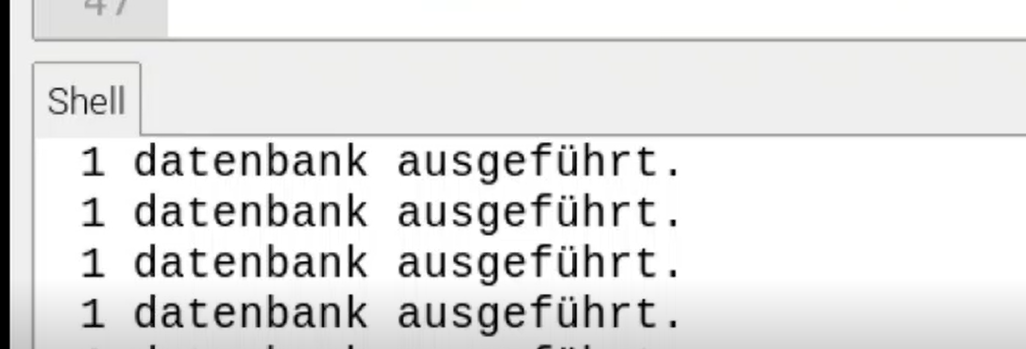
Import mysql.connector

* Notwendig um sich mit der Datenbank zu verbinden



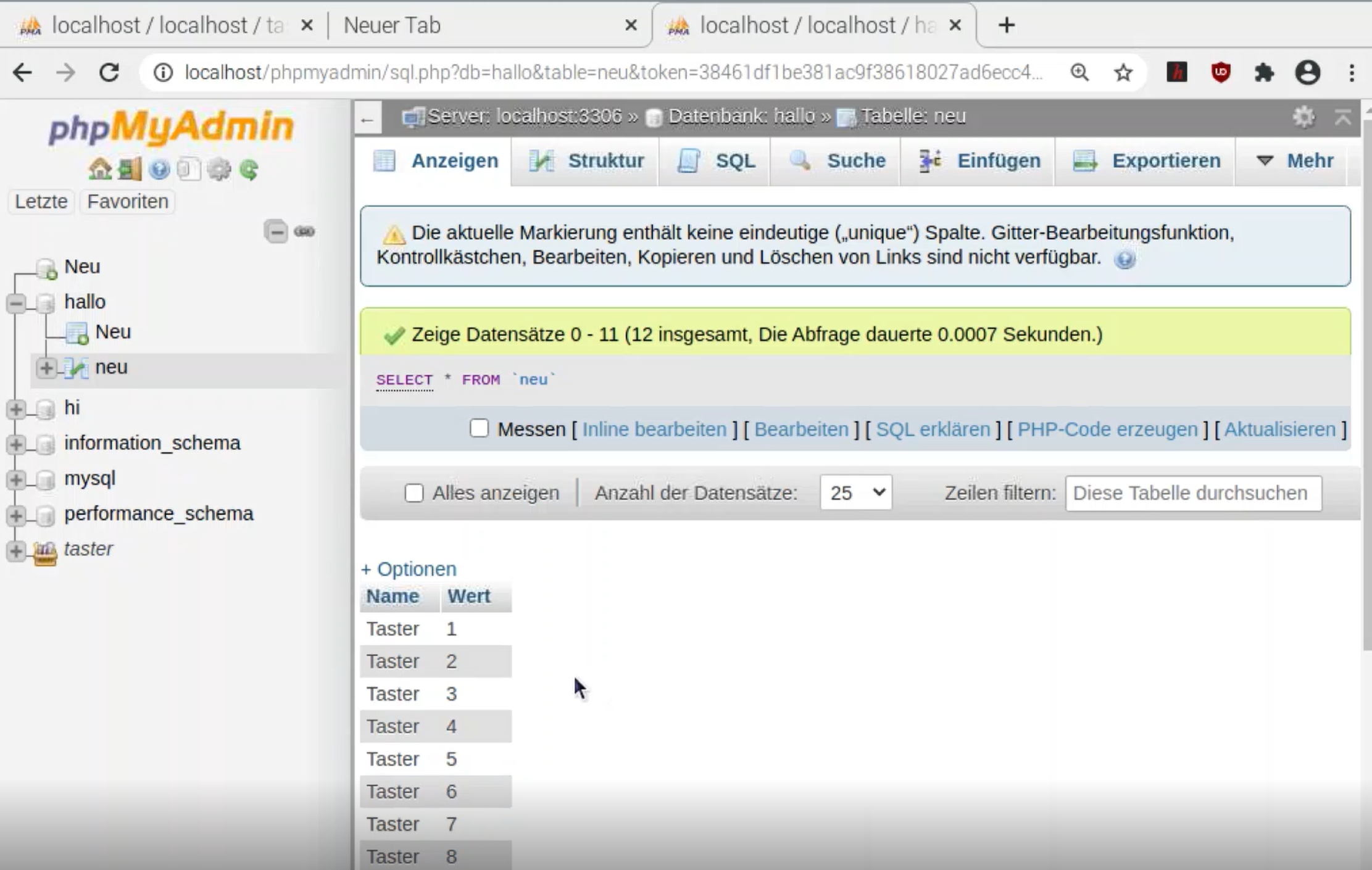
* Wenn Led eingeschaltet
* Datenbank wird eingeloggt
* 1. Host ; 2.Datenbankname ; username; password

Code Teil3

* Wenn Led eingeschaltet Variable zählt hoch
* Die Variable wird somit dann auf der Datenbank gespeichert
* Davor wird in die Spalte Name Taster eingefügt
* Zuletzt bekommt man eine Nachricht Datenbank ausgeführt wenn auf run geklickt wird

Datenbank Ergebnis



* Datenbank Neu
* Variable wird hochgezählt
* Name Taster wird eingefügt