

Tags: [#Abschlussprüfung](#) [#Datenbank](#) [#PHP](#)

# Scriptum

## Wichtige VS Code Erweiterungen:

- PHP Awesome Snippets
- PHP Intelephense
- Ev. Blackbox AI versuchen

## PHP Infos:

- <https://www.php.net/manual/de/intro-what-is.php>
- <https://www.w3schools.com/php/default.asp>

## Verbindungsaufbau zur Datenbank:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbName = "Datenbasename";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbName);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

---

## Vorlage Datenbankdokumentation:

## DB-Dokumentation

### Verwendete Software:

MySQL/MariaDB

Zur Dokumentation: MS Word und PHPMyAdmin als Website auf dem Webserver

### Datenbank:

Das Tabellenschema sowie die Constraints befinden sich als PDF im Anhang.

### Constraints:

Bitte beachten Sie, dass für diese Datenbank bei den Constraints bei ON UPDATE standardmäßig CASCADE verwendet wird.

tbl\_kunden FIDStaat -> tbl\_staat IDStaat 1:n Beziehung weil ein Staat mehreren Kunden zugeordnet werden kann.

ON DELETE RESTRICT weil ein Kunde immer einem Staat angehören muss um die Rechnung oder das Paket verschicken zu können

tbl\_paketversand FIDPaketgroesse -> tbl\_paketgroesse IDPaketgroesse 1:n Beziehung weil eine Größe mehreren Paketen zugeordnet werden kann.

ON DELETE CASCADE sollte eine Paketgröße gelöscht werden sollen auch der Versand gelöscht werden

tbl\_paketversand FIDVersandvariante -> tbl\_versandvarianten IDVersandvariante 1: n Beziehung weil eine Versandvariante mehreren Versendungen zugeordnet werden kann

ON DELETE CASCADE sollte eine Versandvariante gelöscht werden muss zugleich auch der Versand gelöscht werden weil sonst nicht zugeordnet werden kann wie das Paket verschickt werden soll

tbl\_paketversand FIDUStSatz -> tbl\_ustsaetze IDUStSatz 1:n Beziehung weil ein UStSatz mehreren Sendungen zugewiesen werden kann

ON DELETE RESTRICT sollte ein UStSatz gelöscht werden soll nicht automatisch der Versand gelöscht werden weil nur der UStSatz beim Preis wegfällt

tbl\_positionen FIDRechnung -> tbl\_rechnungen IDRechnung 1:n Beziehung weil jeder Rechnung mehrere Positionen zugewiesen werden können ON DELETE CASCADE wenn die Rechnung gelöscht wird kann auch die Position gelöscht werden

tbl\_positionen FIDProdukt -> tbl\_produkte IDProdukt 1:n Beziehung weil mehrere Produkte einer Position zugewiesen werden können

ON DELETE RESTRICT wenn ein Produkt gelöscht werden sollte soll nicht auch die gesamte Position gelöscht werden.

tbl\_produkte FIDUStSatz -> tbl\_ustsaetze IDUStSatz 1:n Beziehung weil ein UStSatz mehreren Produkten zugewiesen werden kann

ON DELET RESTRICT Sollte ein UStSatz gelöscht werden soll nicht auch das Produkt gelöscht werden

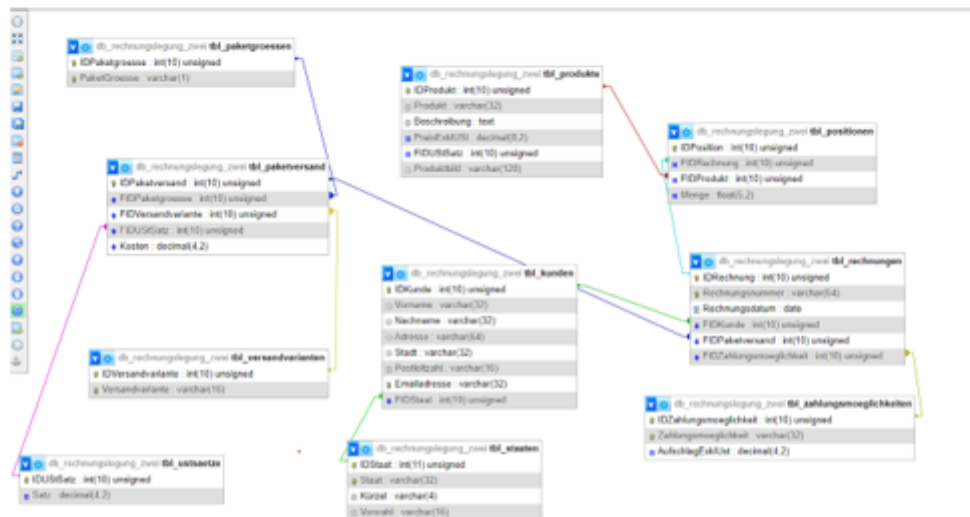
tbl\_rechnungen FIDKunde -> tbl\_kunden IDKunde 1:n Beziehung weil ein Kunde mehrere Rechnungen haben kann

ON DELET RESTRICT sollte ein Kunde gelöscht werden soll nicht auch die Rechnung gelöscht werden

tbl\_rechnungen FIDPaketversand -> tbl\_paketversand IDPaketversand 1:1 Beziehung weil eine Rechnung nur einen Versand haben kann  
ON DELETE RESTRICT sollte ein Paketversand gelöscht werden soll nicht auch die Rechnung gelöscht werden

tbl\_rechnungen FIDZahlungsmoeglichkeit -> tbl\_zahlungsmoeglichkeit IDZahlungsmoeglichkeit 1:1 Beziehung weil eine Rechnung nur eine Zahlungsmoeglichkeit haben kann.  
ON DELETE RESTRICT sollte eine Zahlungsmoeglichkeit gelöscht werden soll nicht die Rechnung gelöscht werden

ER-Diagramm:



DB Dump:

```

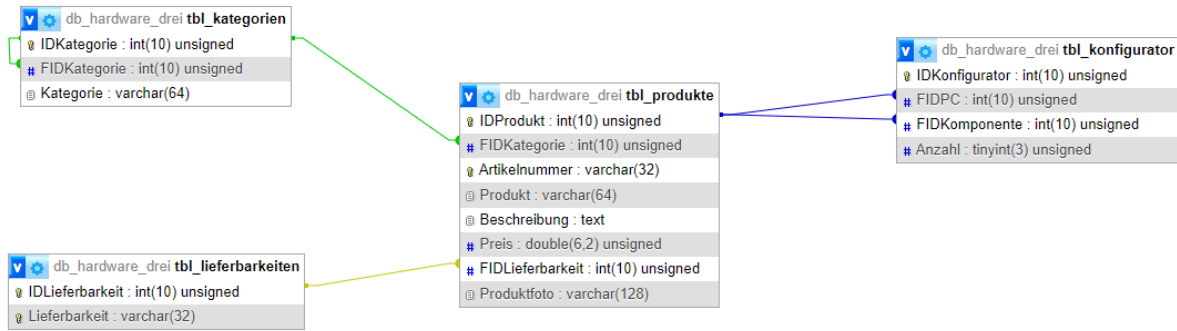
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Erstellungszeit: 29. Jun 2023 um 14:55
-- Server-Version: 10.4.28-MariaDB
-- PHP-Version: 8.0.28

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
    
```

DB-Dump wird außerdem noch als SQL-File angehängt.

ER-Diagramm:



## DB Good To Know:

- Bei einer Recursiven Tabelle verweist die Tabelle auf sich selbst FIDEintrag verweist auf IDEintrag in der tbl\_eintraege.
- Eine n:m Tabelle wird benötigt wenn man eine Zwischentabelle braucht wie z.B: bei der Rechnungslegung. Hier gibt es eine Zwischentabelle (tbl\_positionen).

## Left Join:

```

<body>
  <ul>
    <?php
      $sql="SELECT
        tbl_raume.*,
        tbl_klassen.Bezeichnung
      FROM tbl_raume
      LEFT JOIN tbl_klassen ON tbl_klassen.FIDRaum = tbl_raume.IDRaum
      ORDER BY Raumbezeichnung ASC
      ";
      //left JOIN weil ich alle Räume will auch wenn sie keiner Klasse zugewiesen sind
      $raume = $conn->query($sql) or die ("Fehler in der Query");

      while($raum = $raume->fetch_object()){
        echo('<li>'.$raum->Raumbezeichnung.'</li>');
        $sql="SELECT * FROM tbl_klassen
          WHERE (FIDRaum = $raum->IDRaum )
          ";
        $klassen = $conn->query($sql) or die ("Fehler in der Query bro");
        while($klasse = $klassen->fetch_object()){
          echo('<ul>');
          echo('<li>'.$klasse->Bezeichnung.'</li>');
          echo('</ul>');
        }
      }
    </?php>
  </ul>
</body>
  
```

```
    }  
    ?>  
</ul>  
</body>
```

---

## Recursion:

In der Datenbank verweist die Tabelle auf sich selbst.

Verläuft im Endeffekt immer nach dem gleichen Schema.

```
<?php  
require("includes/common.inc.php");  
require("includes/conn.inc.php");  
function zeigeNewsforum($input = null){  
    global $conn;  
    //es braucht einfach das sql statement mit dem IS NULL statment  
    if(is_null($input)){  
        $sql="SELECT  
            tbl_eintraege.*,  
            tbl_user.Vorname,  
            tbl_user.Nachname  
        FROM tbl_eintraege  
        LEFT JOIN tbl_user ON tbl_user.IDUser = tbl_eintraege.FIDUser  
        WHERE(  
            tbl_eintraege.FIDEintrag IS NULL  
        )  
        ";  
    }  
    else{  
        $sql="SELECT  
            tbl_eintraege.*,  
            tbl_user.Vorname,  
            tbl_user.Nachname  
        FROM tbl_eintraege  
        LEFT JOIN tbl_user ON tbl_user.IDUser = tbl_eintraege.FIDUser  
        WHERE(  
            tbl_eintraege.FIDEintrag = ".$input."  
        )  
        ";  
    }  
    $eintraege = $conn->query($sql) or die ("Fehler in der Query Bro: ".$conn->error."");  
    echo('<ul>');  
    while($eintraege = $eintraege->fetch_object()){  
        echo('<li>'.$eintrag->Vorname.' '.$eintrag->Nachname.' schrieb am: '.date("d.m.y",  
            strtotime($eintrag->Eintragezeitpunkt)).  
            ' um: '.date("H:i:s", strtotime($eintrag->Eintragezeitpunkt)).  
            ':<div>'.$eintrag->Eintrag.'</div>');  
    }  
}
```

```

        //Recursion

        zeigeNewsforum($eintrag->IDEintrag);

        //
        echo('</li>');

    }
    echo('</ul>');
}
?>

<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Newsforum</title>
</head>
<body>
    <h1>Newsforum</h1>
    <?php
    zeigeNewsforum();
    ?>
</body>
</html>

```

## Filter:

Filter wenn nach mehreren Kriterien gefiltert werden soll:

```

<?php
require("includes/common.inc.php");
require("includes/conn.inc.php");
?>

<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Einsatz Mitarbeiter</title>
</head>
<body>
    <h1>Einsätze aller Mitarbeiter</h1>
    <form method="post">
        <label>Vorname Mitarbeiter:</label>
        <input type="text" name="VN_Mitarbeiter">
        <label>Nachname Mitarbeiter:</label>
        <input type="text" name="NN_Mitarbeiter">
        <label>Vorname Kunde:</label>
        <input type="text" name="VN_Kunde">
    
```

```

<label>Nachname Kunde:</label>
<input type="text" name="NNKunde">
<input type="submit" value="FILTERN">
</form>
<ul>
<?php
ta($_POST);
$where = "";
//achtung scheißstift einbauen!!!!
if(count($_POST)>0){
    $arr = [];
    if($_POST["VNMitarbeiter"] != ""){
        $arr[] = "Vorname = '". $_POST["VNMitarbeiter"]."'";
    }
    if($_POST["NNMitarbeiter"] != ""){
        $arr[] = "Nachname='". $_POST["NNMitarbeiter"]."'";
    }
    if(count($arr)>0){
        $where = "WHERE (
            ".implode("AND ", $arr).
        )";
    }
}
$sql = "SELECT * FROM
tbl_Mitarbeiter
".$where."
ORDER BY Nachname ASC, Vorname ASC
";
//ta($sql);
$alleMitarbeiter = $conn->query($sql) or die ("Fehler in der Query Bro: ".$conn-
>error."");
while($mitarbeiter = $alleMitarbeiter->fetch_object()){
    echo('<li><b>'.$mitarbeiter->Vorname.' '.$mitarbeiter->Nachname.'</b></li>');
    //alarm FIDMitarbeiter nicht FIDKunde
    if(count($_POST)>0){
        $arrZwei = ["FIDMitarbeiter = '". $mitarbeiter->IDMitarbeiter."'"];
        if($_POST["VNKunde"] != ""){
            $arrZwei[] = "Vorname= '". $_POST["VNKunde"]."'";
        }
        if($_POST["NNKunde"] != ""){
            $arrZwei[] = "Nachname = '". $_POST["NNKunde"]."'";
        }
    }
    $sql = "SELECT
tbl_einsatz.*,
tbl_kunden.Vorname,
tbl_kunden.Nachname
FROM tbl_einsatz
INNER JOIN tbl_kunden ON tbl_kunden.IDKunde = tbl_einsatz.FIDKunde

```

```

        WHERE (
            ".implode("AND ", $arrZwei)."
        )
    ";
    //ta($sql);
    $einsaetze = $conn->query($sql) or die ("ALARM FEHLER: ".$conn->error."");
    while($einsatz = $einsaetze->fetch_object()){
        echo('<ul>');
        echo('<li>Eingesetzt bei: '.$einsatz->Vorname.' '.$einsatz->Nachname.' von:
'.date("d-m-Y H:i:s", strtotime($einsatz->Startzeitpunkt)).' bis: '.date("d-m-Y H:i:s",
strtotime($einsatz->Endzeitpunkt)).'</li>');
        echo('</ul>');
    }
}
?>
</ul>
</body>
</html>

```

## Weiterleitung (Bsp.: Hardware):

```

<?php
require("includes/common.inc.php")
require("includes/conn.inc.php");

function zeigeKategorien($input = null){
    global $conn;
    if(is_null($input)){
        $sql = "SELECT * FROM
tbl_kategorien
WHERE (
    FIDKategorie IS NULL
)
";
    }
    else{
        $sql = "SELECT * FROM
tbl_kategorien
WHERE (
    FIDKategorie = $input
)
";
    }
    $kategorien = $conn->query($sql) or die ("Fehler in der Query:".$conn->error."");
    echo('<ul>');
    while($kategorie = $kategorien->fetch_object()){
        echo('<li>');
        echo('<a href="produkteKategorien.php?idKategorie='.$kategorie->IDKategorie.'">'.$kategorie->Kategorie.'</a>');
    }
}

```



```

        //Rekursion
        zeigeKategorien($kategorie->IDKategorie);
        echo('</li>');
    }
    echo('</ul>');
}
?>

<!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Alle Kategorien</title>
</head>
<body>
    <h1>
        Alle Kategorien
    </h1>
    <?php
        zeigeKategorien();
    ?>
</body>
</html>

```

Zweite Seite die durch die Weiterleitung von unserem Anker passieren muss:

```

<?php
require("includes/common.inc.php");
require("includes/conn.inc.php");

ta($_GET["idKategorie"]); //zieht sich die idKategorie aus der URL!
$id = 0;
if(count($_GET) > 0 && $_GET["idKategorie"] != ""){
    $id = $_GET["idKategorie"];
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Produkte</title>
</head>
<body>
    <ul>
    <?php
        if($id > 0){
            $sql = "SELECT
                tbl_produkte.*,

```

```

        tbl_lieferbarkeiten.Lieferbarkeit
    FROM tbl_produkte
    INNER JOIN tbl_lieferbarkeiten ON tbl_lieferbarkeiten.IDLieferbarkeit =
tbl_produkte.FIDLieferbarkeit
    WHERE(
        FIDKategorie = $id
    )
";
    ta($sql);
    $produkte = $conn->query($sql) or die ("Fehler in der Query meister".$conn-
>error."");
    while($produkt = $produkte->fetch_object()){
        if(is_null($produkt->Produktfoto)){
            $produktfoto = "Leider ist kein Produktfoto verfügbar";
        }
        else{
            $produktfoto = 'Produkt.'">';
        }
        echo('
<li>'.$produkt->Produkt.'('.$produkt->Artikelnummer.')
<div>'.$produkt->Beschreibung.'</div>
Produktpreis: '.$produkt->Preis.'€ ('.$produkt->Lieferbarkeit.')
'.$produktfoto.'
</li>
');
    }
}
?>
</ul>
</body>
</html>

```

## Ausgabe mit Count (Hoffentlich kommt das nicht):

```

<?php
require("includes/config.inc.php");
require("includes/common.inc.php");
require("includes/conn.inc.php");
ta($_POST);
?>
<!doctype html>
<html lang="de">
    <head>
        <title>Hardwarekomponenten</title>
        <meta charset="utf-8">
        <link rel="stylesheet" href="css/common.css">
    </head>
    <body>

```

```
<h1>Hardwarekomponenten</h1>
<nav>
    <ul>
        <li><a href="index.html">zur Startseite</a></li>
        <li><a href="kategorien.php">Kategorien</a></li>
    </ul>
</nav>
<form method="post">
    <label>
        Komponente:
        <input type="text" name="K">
    </label>
    <label>
        Artikelnummer:
        <input type="text" name="ANr">
    </label>
    <input type="submit" value="filtern">
</form>
<ul>
<?php
    $sql = "
        SELECT
            *
        FROM tbl_produkte
        WHERE(
            FIDKategorie=2 //weil FIDKategorien 2 die IndividualPCs sind
        )
    ";
    $pcs = $conn->query($sql) or die("Fehler in der Query: " . $conn->error . "<br>"
. $sql);
    while($pc = $pcs->fetch_object()) {
        $anzKomponenten = 1;
        if(count($_POST)>0) {
            $arr = ["tbl_konfigurator.FIDPC=" . $pc->IDProdukt];
            if(strlen($_POST["K"])>0) {
                $arr[] = "Produkt LIKE '%" . $_POST["K"] . "%'";
            }
            if(strlen($_POST["ANr"])>0) {
                $arr[] = "Artikelnummer LIKE '%" . $_POST["ANr"] . "%'";
            }
            if(count($arr)>1) {
                $sql = "
                    SELECT
                        COUNT(*) AS cnt
                    FROM tbl_konfigurator
                    INNER JOIN tbl_produkte ON
tbl_produkte.IDProdukt=tbl_konfigurator.FIDKomponente
                    WHERE(
                        " . implode(" AND ", $arr) . "
                ";
            }
        }
    }
}
```

```

        )
        ";
        $koms = $conn->query($sql) or die("Fehler in der Query: " . $conn->error . "<br>" . $sql);
        $komp = $koms->fetch_object();
        $anzKomponenten = $komp->cnt;
        ta($komp);
    }
}
if($anzKomponenten>0) {
    if(is_null($pc->Produktfoto)) {
        $bild = "";
    }
    else {
        $bild = 'Produkt .
' ">';
    }
    echo('
        <li>
            ' . $pc->Produkt . ' (' . $pc->Artikelnummer . '; ' . $pc->IDProdukt . ')
            <div>' . $pc->Beschreibung . '</div>
            <ul>
                ');
    // ---- alle Komponenten des ausgewählten PCs: ----
    $sql = "
        SELECT
            tbl_konfigurator.Anzahl,
            tbl_produkte.*,
            tbl_lieferbarkeiten.Lieferbarkeit
        FROM tbl_konfigurator
        INNER JOIN tbl_produkte ON
tbl_produkte.IDProdukt=tbl_konfigurator.FIDKomponente
        INNER JOIN tbl_lieferbarkeiten ON
tbl_lieferbarkeiten.IDLieferbarkeit=tbl_produkte.FIDLieferbarkeit
        WHERE(
            tbl_konfigurator.FIDPC=" . $pc->IDProdukt . "
        )
    ";
    //ta($sql);
    $komponenten = $conn->query($sql) or die("Fehler in der Query: " .
$conn->error . "<br>" . $sql);
    $sum = 0;
    while($k = $komponenten->fetch_object()) {
        $sum = $sum + $k->Anzahl*$k->Preis;
        if(is_null($k->Produktfoto)) {
            $bildk = "";
        }
        else {

```

```

        $bildk = 'Produkt . '">';
    }
    echo('
        <li>
            ' . $k->Produkt . ' (' . $k->Artikelnummer . ')
            <div>' . $k->Beschreibung . '</div>
            EUR ' . $k->Preis . ' (' . $k->Lieferbarkeit . ')
            ' . $bildk . '
        </li>
    ');
}
// -----
echo('
    </ul>
    EUR ' . $sum . '
    ' . $bild . '
</li>
');
}
}
?>
</ul>
</body>
</html>

```