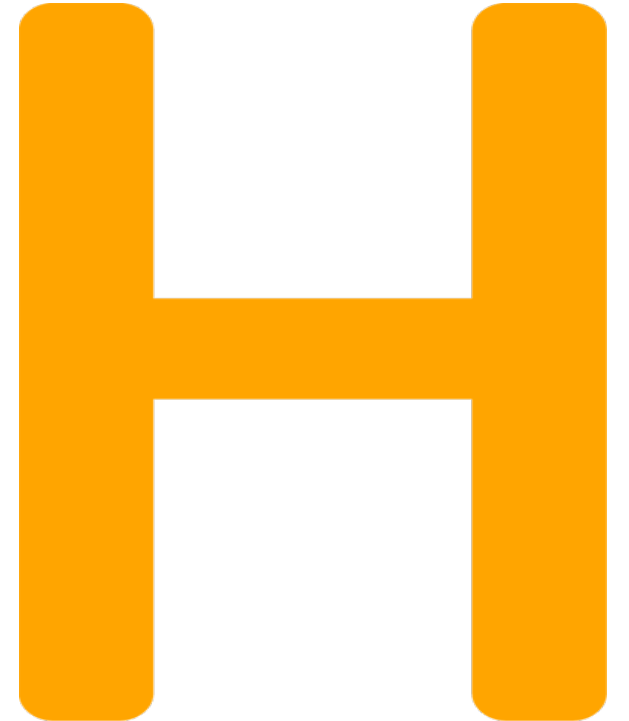


# Refugeeks / IntoCode

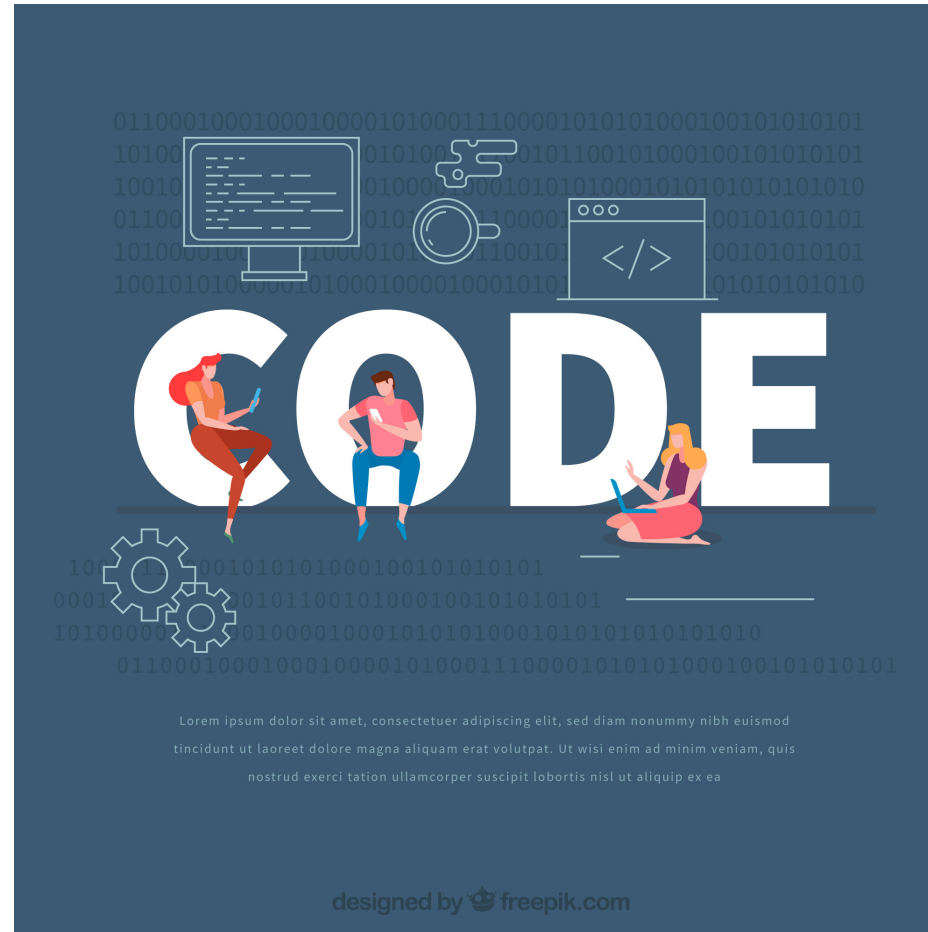
## Dynamische Websites

**22.09.2020 / 01:**

Datenbank-Operationen (Select, Update, Delete)



# Dynamische Websites – Datenbank - Select



# Dynamische Website

## Datenbank-Anbindung

### Anforderung:

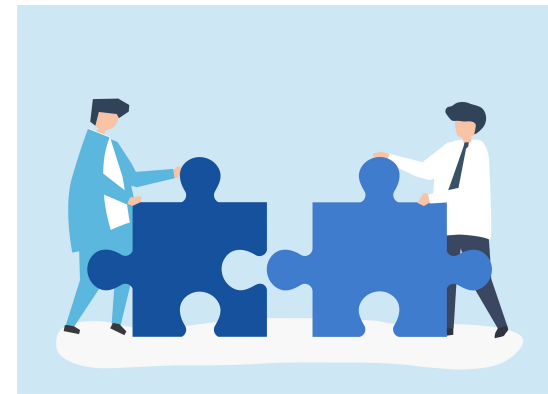
Es sollen alle Mitarbeiter aus der Datenbank user\_repository abgefragt und tabellarisch auf der Website ausgegeben werden

### IST-Stand:

- Datenbankverbindung vorhanden
- Keine Datenbank-Tabelle vorhanden
- Kein Query vorhanden
- Keine Tabelle zur Ausgabe vorhanden

### Lösungsansatz:


- Datenbank-Tabelle über SQL-Query in phpMyAdmin erstellen
- Query erstellen
- PHP-Funktion zur Abfrage verwenden
- HTML-Table benutzen und Werte dynamisch generieren



# Aufgabe

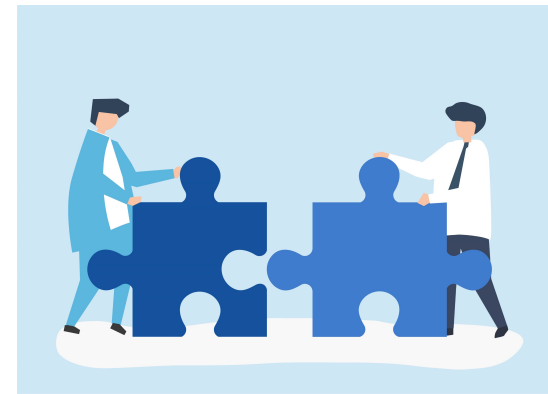
## Datenbankaufbau

Baue die Tabelle wie im Bild unten zu sehen auf. Verwende hierfür eine selbstgeschriebene SQL-Anweisung.

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra
1	ID 	int(11)			Nein	<i>kein(e)</i>		AUTO_INCREMENT
2	FORENAME	varchar(255)	utf8_general_ci		Nein	<i>kein(e)</i>		
3	SURENAME	varchar(255)	utf8_general_ci		Nein	<i>kein(e)</i>		
4	AGE	int(11)			Nein	<i>kein(e)</i>		
5	POSITION	varchar(255)	utf8_general_ci		Nein	<i>kein(e)</i>		
6	ACTIVE	tinyint(1)			Nein	<i>kein(e)</i>		
7	ROLE_ID	int(11)			Nein	<i>kein(e)</i>		

# Dynamische Website

## Einführung: Select – PHP Datenbank



1. Abfrage definieren und in Variable speichern
2. Das Verbindungsobjekt(\$conn) stellt eine Methode bereit, um die Abfrage an die Datenbank zu senden  
->**query()**
3. Das Ergebnis der Abfrage in einer Variable speichern
4. Ergebnisanzahl prüfen (mind. 1)
5. Methode ->**fetch\_assoc()** nutzen um die Spalten-Ergebnisse in einem assoziativen Array zu speichern
6. Schleife nutzen um alle Zeilen-Ergebnisse abzuarbeiten
7. Ergebnisse verarbeiten

```
<?php

$query = "SELECT * FROM EMPLOYEE e";

$result = $conn->query($query);

if($result->num_rows > 0){
    while($row = $result->fetch_assoc()){
        echo "id : ".$row['ID'];
        echo "<br>";
        echo "Vorname : ".$row['FORENAME'];
        echo "<br>";
    }
}

?>
```



# Dynamische Website

**Select – PHP**      **Datenbank - Mit Tabellenausgabe**

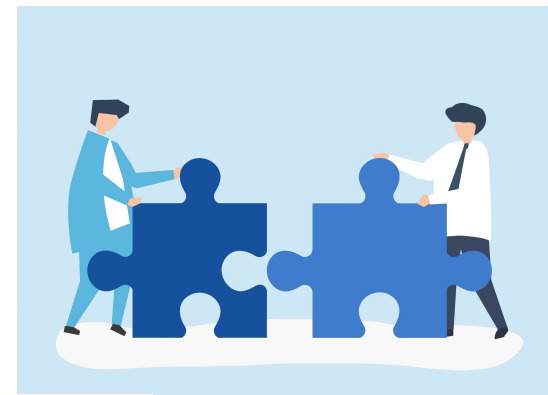
```
<table>
  <tr>
    <th>ID</th>
    <th>Vorname</th>
    <th>Nachname</th>
    <th>Alter</th>
    <th>Position</th>
    <th>Aktiv</th>
    <th>Role_ID</th>
  </tr>
```

```
<?php
    $query = "SELECT * FROM EMPLOYEE e";

    $result = $conn->query($query);

    if($result->num_rows > 0){
        while($row = $result->fetch_assoc()){
            echo "<tr>";
            echo "<td>".$row['ID']. "</td>";
            echo "<td>".$row['FORENAME']. "</td>";
            echo "<td>".$row['SURENAME']. "</td>";
            echo "<td>".$row['AGE']. "</td>";
            echo "<td>".$row['POSITION']. "</td>";
            echo "<td>".$row['ACTIVE']. "</td>";
            echo "<td>".$row['ROLE_ID']. "</td>";
            echo "</tr>";
        }
    }
}
```

```
?>
</table>
```



# Dynamische Website

Select – PHP      Datenbank

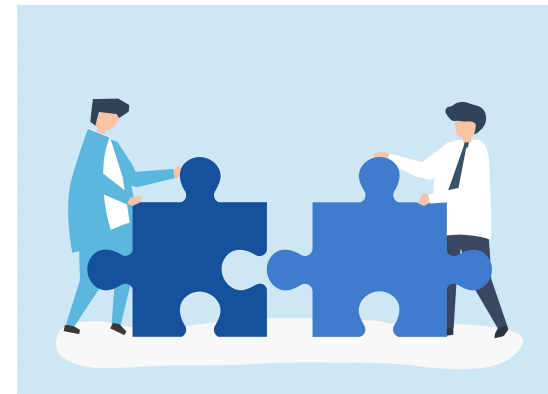
Weitere Methoden für Ergebniszuzuweisung:

Indexiertes Array:

```
$result = $conn->fetch_row();
```

Objekt:

```
$result = $conn->fetch_object();
```



# Dynamische Websites – Datenbank – Insert/Update/Delete





# Dynamische Website

## Datenbank-Anbindung

### Anforderung:

Es soll ein Formular zur Verfügung gestellt werden, mit dem neue Mitarbeiter angelegt/bearbeitet und gelöscht werden können

### IST-Stand:

- Datenbankverbindung vorhanden
- Kein Formular vorhanden

### Lösungsansatz:

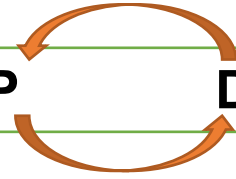
- SQL-Statement erstellen
- PHP-Funktionen für SQL-Statements verwenden



# Dynamische Website

Einführung: Insert – PHP

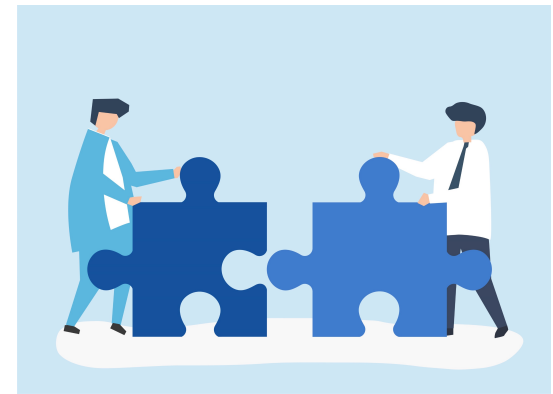
Datenbank



1. HTML-Formular anlegen
2. Datensatz eintragen: SQL-Statement verfassen
3. Methode query(); nutzen
4. Prüfung ob das Statement erfolgreich ausgeführt wurde

(Ausschnitt)

```
$query = "INSERT INTO EMPLOYEE(forename, surname, age,  
  
    if ($conn->query($query) === TRUE) {  
        echo "<h4>Mitarbeiter erfolgreich angelegt</h4>";  
    } else {  
        echo "Error: " . $query . "<br>" . $conn->error;  
    }  
}
```



# Dynamische Website

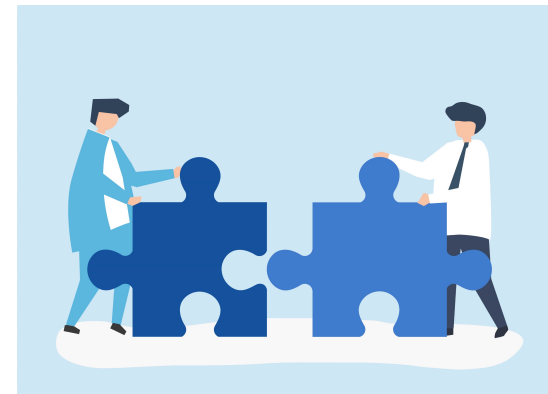
Einführung: Update – PHP

Datenbank

1. HTML-Formular anlegen
2. Datensatz bearbeiten: SQL-Statement verfassen
3. Methode query(); nutzen
4. Prüfung ob das Statement erfolgreich ausgeführt wurde

(Ausschnitt)

```
$query = "UPDATE EMPLOYEE set forename = '". $forename . "'",  
if ($conn->query($query) === TRUE) {  
    echo "<h4>Mitarbeiter erfolgreich aktualisiert</h4>";  
} else {  
    echo "Error: " . $query . "<br>" . $conn->error;  
}
```



# Dynamische Website

Einführung: Delete – PHP  Datenbank

1. HTML-Formular anlegen
2. Datensatz löschen: SQL-Statement verfassen
3. Methode query(); nutzen
4. Prüfung ob das Statement erfolgreich ausgeführt wurde

(Ausschnitt)

```
$query = "DELETE FROM EMPLOYEE WHERE ID = ('".$employee."')";  
  
if ($conn->query($query) === TRUE) {  
    echo "<h4>Mitarbeiter erfolgreich entfernt</h4>";  
} else {  
    echo "Error: " . $query . "<br>" . $conn->error;  
}
```



# Finish

# Dynamische Websites

# PHP VI





<https://www.w3schools.com/php/default.asp>

<https://www.php.net/manual/de/langref.php>

PHP

<https://www.w3schools.com/tags/default.asp>

HTML

<https://www.w3schools.com/cssref/default.asp>

CSS

