

Group Id 905

-Praktikum DBWT. Autoren :

-Fachrial Dimas Putra, Perdana, 3503937

-Jericho, Jordan, 3536333

Meilenstein	Aufgabe	Geschätzte Zeit (in Minuten)	Tatsächlich benötigte Zeit (in Minuten)
1	1	3	1
	2	1	1
	3	-	-
	4	30	50
		Die Aufgabe war komplizierter als wir erwartet haben. Wir müssten mehr Attributes neu lernen.	
	5	30	25
	6	30	30
	7	5	5
	8	15	15
	9	60	50
	10	120	145
		Wir waren ein bisschen verwirrend, um die Objekte und Elemente zu platzieren.	

Aufgabe 6.1

1. Accept → enthält den Inhaltstyp, den der Benutzer verstehen kann.
2. Accept-Encoding → enthält die Inhaltscodierung, die der Benutzer verstehen kann.
3. Accept-Language → enthält die natürliche Sprache und Gebietsschema, das der Kunde bevorzugt.
4. Cache-Control → enthält Anweisungen, die die Zwischenspeicherung in Browsern und gemeinsam genutzten Caches (z. B. Proxies, CDNs) steuern.

5. Connection → steuert, ob die Netzverbindung nach Beendigung die aktuelle Benutzung offen bleibt. (*keep-alive*)

Aufgabe 6.2

1. sec-ch-ua → das Branding des Benutzeragenten und wichtige Versionsinformationen bereitstellen.
2. sec-ch-ua-mobile → anzuzeigen, ob sich der Browser auf einem Mobilgerät befindet.
3. sec-ch-ua-platform → die Plattform oder das Betriebssystem bereitstellen, auf dem der Benutzeragent ausgeführt wird.

da gibt es nicht so viele Anfrage wie in der Webseite der Bibliothek.

Aufgabe 9.2

- background-color
- border
- border-bottom
- border-left
- color
- display
- font-family
- grid-auto-flow
- grid-template-columns
- justify-content
- line-height
- list-style-type
- margin
- margin-bottom
- max-width
- padding
- text-decoration
- text-warp
- word-break

Quelle:

https://de.wikipedia.org/wiki/Hypertext_Markup_Language

<https://developer.mozilla.org/en-US/>

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

<https://www.loremipsum.de/>

Meilenstein	Aufgabe	Geschätzte Zeit (in Minuten)	Tatsächlich benötigte Zeit (in Minuten)
2	1	3	2
	2	30	25
	3	120	240++
		Wir sind immer noch mit der Bearbeitung von den Get-Parametern verwirrt.	
	4	4	4
	5	120	240
		Wir haben PHP noch nicht gewöhnlich verwendet	
	6	120	120
	7	120	120
	8	180	151

Aufgabe 3.1

Allergensliste fehlt komma(,) nach 'Fisch'

Meal → Allergen fehlt eckige Klammer (']') nach '13'

fehlt noch ein Klammer (')') ins Ende des ersten 'if'

fehlt 'function' am Anfang des Funktion `calcMeanStars`

Aufgabe 3.2

dem Nutzer ermöglichen die Bewertungstabelle nach der minimale Sterne oder einem Suchwort zu filtern (TOP und FLOP kann nach der Aufgabenstellung auch als Filter verwendet werden), Sprache wechseln, Beschreibung des Gerichtes zu ein/ausblenden.

Aufgabe 3.3

echo : Ausgabe-Funktion in PHP.
const : Konstante Variable
?var : Get Variable/Parameter
Float : Datentyp für Dezimalzahlen
\$var : Variable
\$Var =[] : Array
foreach : Array iterator
if/else : Verzweigung.
isset : Variable existiert
empty : Variablenwert nicht gleich Null

Meilenstein	Aufgabe	Geschätzte Zeit (in Minuten)	Tatsächlich benötigte Zeit (in Minuten)
3	1	10	8
	2	30	20
	3	45	40
	4	4	4
	5	60	45
	6	120	120
	7	45	45
	8	180	151
	9	120	110
	10	10	10
	11	90	90

Aufgabe 3

1) Alle Daten aller Gerichte

SELECT * FROM gericht								
id	name	beschreibung	erfasst_am	vegetarisch	vegan	preis_intern	preis_extern	
1	1 Bratkartoffeln mit Speck u...	Kartoffeln mit Zwiebeln un...	2020-08-25	0	0	2.3	4	
2	3 Bratkartoffeln mit Zwiebeln	Kartoffeln mit Zwiebeln un...	2020-08-25	1	1	2.3	4	
3	4 Grilltofu	Fein gewürzt und mariniert	2020-08-25	1	1	2.5	4.5	
4	5 Lasagne	Klassisch mit Bolognesesoß...	2020-08-24	0	0	2.5	4.5	
5	6 Lasagne vegetarisch	Klassisch mit Sojagranulat...	2020-08-24	1	0	2.5	4.5	
6	7 Hackbraten	Nicht nur für Hacker	2020-08-25	0	0	2.5	4	
7	8 Gemüsepfanne	Gesundes aus der Region, d...	2020-08-25	1	1	2.3	4	
8	9 Hühnersuppe	Suppenhuhn trifft Petersil...	2020-08-25	0	0	2	3.5	
9	10 Forellenfilet	mit Kartoffeln und Dill dip	2020-08-22	0	0	3.8	5	
10	11 Kartoffel-Lauch-Suppe	der Klassische Bauchwärmer...	2020-08-22	1	0	2	3	
11	12 Kassler mit Rosmarinkartof...	dazu Salat und Senf	2020-08-23	0	0	3.8	5.2	
12	13 Drei Reibekuchen mit Apfel...	grob geriebene Kartoffeln ...	2020-08-23	1	0	2.5	4.5	
13	14 Pilzpfanne	die legendäre Pfanne aus P...	2020-08-23	1	0	3	5	

2) Das Erfassungsdatum aller Gerichte

```
SELECT erfasst_am FROM gericht
```

	erfasst_am
1	2020-08-25
2	2020-08-25
3	2020-08-25
4	2020-08-24
5	2020-08-24
6	2020-08-25
7	2020-08-25
8	2020-08-25
9	2020-08-22
10	2020-08-22
11	2020-08-23
12	2020-08-23
13	2020-08-23

3) Das Erfassungsdatum sowie den Namen (als Attributname Gerichtname) aller Gerichte absteigend sortiert nach Gerichtname.

```
SELECT name,erfasst_am FROM gericht ORDER BY name desc
```

	name	erfasst_am
1	Tomatenbrötchen	2020-08-25
2	Suppenkreation á la Chef	2020-08-26
3	Schinkenbrötchen	2020-08-25
4	Pilzpfanne vegan	2020-08-24
5	Pilzpfanne	2020-08-23
6	Mousse au Chocolat	2020-08-26
7	Lasagne vegetarisch	2020-08-24
8	Lasagne	2020-08-24
9	Kassler mit Rosmarinkartoffeln	2020-08-23
10	Käsebrötchen	2020-08-24
11	Kartoffel-Lauch-Suppe	2020-08-22
12	Hühnersuppe	2020-08-25
13	Hackbraten	2020-08-25

4) Den Namen sowie die Beschreibung der Gerichte aufsteigend sortiert nach Name, wobei nur 5 Datensätze dargestellt werden sollen.

```
SELECT name,beschreibung FROM gericht ORDER BY name asc LIMIT 5
```

	name	beschreibung
1	Bratkartoffeln mit Speck und Zwiebeln	Kartoffeln mit Zwiebeln und gut Speck
2	Bratkartoffeln mit Zwiebeln	Kartoffeln mit Zwiebeln und ohne Speck
3	Drei Reibekuchen mit Apfelmus	grob geriebene Kartoffeln aus der Region
4	Forellenfilet	mit Kartoffeln und Dilldip
5	Gemüsepfanne	Gesundes aus der Region, deftig angebraten

- 5) Ändern Sie die vorherige Abfrage so ab, so dass 10 Datensätze dargestellt werden, die nach den ersten 5 Datensätzen folgen. (Die ersten 5 Datensätze werden übersprungen)

```
SELECT name,beschreibung FROM gericht ORDER BY name asc LIMIT 5 OFFSET 5
```

	name	beschreibung
1	Grilltofu	Fein gewürzt und mariniert
2	Hackbraten	Nicht nur für Hacker
3	Hühnersuppe	Suppenhuhn trifft Petersilie
4	Kartoffel-Lauch-Suppe	der klassische Bauchwärmer mit frischen Kräutern
5	Käsebrötchen	schmeckt vor und nach dem Essen

- 6) Zeigen Sie alle möglichen Allergen-Typen (typ), wobei Sie keine doppelten Einträge darstellen.

```
SELECT DISTINCT typ FROM allergen
```

	typ
1	Getreide (Gluten)
2	Allergen

- 7) Namen von Gerichten, deren Name mit einem „K“ beginnt.

```
SELECT name FROM gericht WHERE name LIKE 'K%'
```

	name
1	Kartoffel-Lauch-Suppe
2	Kassler mit Rosmarinkartoffeln
3	Käsebrötchen

- 8) Ids und Namen von Gerichten, deren Namen ein „suppe“ an beliebiger Stelle enthält.

```
1 ✓ SELECT id,name FROM gericht WHERE name LIKE '%suppe%'
```

	id	name
1	9	Hühnersuppe
2	11	Kartoffel-Lauch-Suppe
3	20	Suppenkreation á la Chef

9) Alle Kategorien, die keine Elterneinträge besitzen.

1	✓	<code>SELECT * FROM kategorie WHERE eltern_id IS NULL</code>			
	✗	<div> <div>2 rows</div> <div> <div>id</div> <div>name</div> <div>eltern_id</div> <div>bildname</div> </div> </div>			
		1	Aktionen	<null>	kat_aktionen.png
		2	Menus	<null>	kat_menu.gif

10) Korrigieren Sie den Wert „Dinkel“ in der Tabelle allergen mit dem code a6 zu „Kamut“.

1	✓	<code>SELECT * FROM allergen;</code>			
2	✗	<code>UPDATE allergen SET name = 'Kamut' WHERE name = 'Dinkel' && code = 'a6';</code>			
		<div> <div>21 rows</div> <div> <div>code</div> <div>name</div> <div>typ</div> </div> </div>			
		1	a	Getreideprodukte	Getreide (Gluten)
		2	a1	Weizen	Allergen
		3	a2	Roggen	Allergen
		4	a3	Gerste	Allergen
		5	a4	Dinkel	Allergen
		6	a5	Hafer	Allergen
		7	a6	Kamut	Allergen
		8	b	Fisch	Allergen
		9	c	Krebstiere	Allergen
		10	d	Schwefeldioxid/Sulfit	Allergen
		11	e	Sellerie	Allergen
		12	f	Milch und Laktose	Allergen
		13	f1	Butter	Allergen

11) Fügen Sie das Gericht „Currywurst mit Pommes“ hinzu und tragen Sie es in der Kategorie „Hauptspeise“ (3) ein.

```
INSERT INTO gericht
(id,name,beschreibung,erfasst_am,vegetarisch,vegan,preis_intern,preis_extern)
VALUES (21,'Currywurst mit Pommes','ganz lecker','2022-11-15',0,0,3.5,5);
```

11	12	Kassler mit Rosmarinkartoffeln	dazu Salat und Senf	2020-08-23	0	0	
12	13	Drei Reibekuchen mit Apfelmus	grob geriebene Kartoffeln aus der Region	2020-08-23	1	0	
13	14	Pilzpfanne	die legendäre Pfanne aus Pilzen der Saison	2020-08-23	1	0	
14	15	Pilzpfanne vegan	die legendäre Pfanne aus Pilzen der Saison ohne Käse	2020-08-24	1	1	
15	16	Käsebrötchen	schmeckt vor und nach dem Essen	2020-08-24	1	0	
16	17	Schinkenbrötchen	schmeckt auch ohne Hunger	2020-08-25	0	0	
17	18	Tomatenbrötchen	mit Schnittlauch und Zwiebeln	2020-08-25	1	1	
18	19	Mousse au Chocolat	sahnige schweizer Schokolade rundet jedes Essen ab	2020-08-26	1	0	
19	20	Suppenkreation á la Chef	was verschafft werden muss, gut und günstig	2020-08-26	0	0	
20	21	Currywurst mit Pommes	ganz lecker	2022-11-15	0	0	

Aufgabe 6

1) Alle Gerichte mit allen zugehörigen Allergenen.

```
SELECT id,ger.name,g.code,al.name,al.typ FROM gericht ger JOIN
gericht_hat_allergen g ON ger.id = g.gericht_id JOIN allergen al on g.code
= al.code
```

	id	ger.name	code	al.name	typ
1	1	Bratkartoffeln mit Speck und Zwiebeln	h	Nüsse	Allergen
2	1	Bratkartoffeln mit Speck und Zwiebeln	a3	Gerste	Allergen
3	1	Bratkartoffeln mit Speck und Zwiebeln	a4	Dinkel	Allergen
4	1	Bratkartoffeln mit Speck und Zwiebeln	c	Krebstiere	Allergen
5	1	Bratkartoffeln mit Speck und Zwiebeln	f1	Butter	Allergen
6	3	Bratkartoffeln mit Zwiebeln	f1	Butter	Allergen
7	3	Bratkartoffeln mit Zwiebeln	a6	Dinkel	Allergen
8	3	Bratkartoffeln mit Zwiebeln	i	Erdnüsse	Allergen
9	10	Forellenfilet	h3	Walnüsse	Allergen
10	10	Forellenfilet	d	Schwefeldioxid/Sulfit	Allergen
11	10	Forellenfilet	f	Milch und Laktose	Allergen
12	8	Gemüsepfanne	a3	Gerste	Allergen
13	4	Grilltofu	a3	Gerste	Allergen
14	4	Grilltofu	f1	Butter	Allergen
15	4	Grilltofu	a4	Dinkel	Allergen
16	4	Grilltofu	h3	Walnüsse	Allergen
17	7	Hackbraten	h1	Mandeln	Allergen
18	7	Hackbraten	a2	Roggen	Allergen
19	7	Hackbraten	h3	Walnüsse	Allergen
20	7	Hackbraten	c	Krebstiere	Allergen
21	9	Hühnersuppe	a2	Roggen	Allergen
22	12	Kassler mit Rosmarinkartoffeln	f2	Käse	Allergen
23	12	Kassler mit Rosmarinkartoffeln	h1	Mandeln	Allergen
24	12	Kassler mit Rosmarinkartoffeln	a5	Hafer	Allergen

2) Ändern Sie die vorherige Abfrage so ab, dass alle existierenden Gerichte dargestellt werden (auch wenn keine Allergene enthalten sind).

```
SELECT id,ger.name,g.code,al.name,al.typ FROM gericht ger LEFT JOIN
gericht_hat_allergen g ON ger.id = g.gericht_id LEFT JOIN allergen al on
g.code = al.code
```

	id	ger.name	code	al.name	typ
1	1	Bratkartoffeln mit Speck und Zwiebeln	h	Nüsse	Allergen
2	1	Bratkartoffeln mit Speck und Zwiebeln	a3	Gerste	Allergen
3	1	Bratkartoffeln mit Speck und Zwiebeln	a4	Dinkel	Allergen
4	1	Bratkartoffeln mit Speck und Zwiebeln	c	Krebstiere	Allergen
5	1	Bratkartoffeln mit Speck und Zwiebeln	f1	Butter	Allergen
6	3	Bratkartoffeln mit Zwiebeln	f1	Butter	Allergen
7	3	Bratkartoffeln mit Zwiebeln	a6	Dinkel	Allergen
8	3	Bratkartoffeln mit Zwiebeln	i	Erdnüsse	Allergen
9	13	Drei Reibekuchen mit Apfelmus	<null>	<null>	<null>
10	10	Forellenfilet	h3	Walnüsse	Allergen
11	10	Forellenfilet	d	Schwefeldioxid/Sulfit	Allergen
12	10	Forellenfilet	f	Milch und Laktose	Allergen
13	8	Gemüsepfanne	a3	Gerste	Allergen
14	4	Grilltofu	a3	Gerste	Allergen
15	4	Grilltofu	f1	Butter	Allergen
16	4	Grilltofu	a4	Dinkel	Allergen
17	4	Grilltofu	h3	Walnüsse	Allergen
18	7	Hackbraten	h1	Mandeln	Allergen
19	7	Hackbraten	a2	Roggen	Allergen
20	7	Hackbraten	h3	Walnüsse	Allergen
21	7	Hackbraten	c	Krebstiere	Allergen
22	9	Hühnersuppe	a2	Roggen	Allergen
23	11	Kartoffel-Lauch-Suppe	<null>	<null>	<null>
24	16	Käsebrötchen	<null>	<null>	<null>
25	12	Kassler mit Rosmarinkartoffeln	f2	Käse	Allergen

3) Ändern Sie die vorherige Abfrage so ab, so dass im Ergebnis alle existierenden Allergene dargestellt werden (auch wenn diese nicht einem Gericht zugeordnet sind).

```
SELECT id,ger.name,g.code,al.name,al.typ FROM gericht ger JOIN
gericht_hat_allergen g ON ger.id = g.gericht_id RIGHT JOIN allergen al on
g.code = al.code
```

1	<null>	<null>	<null>	Getreideprodukte	Getreide (Gluten)
2	15 Pilzpfanne vegan	a1	Weizen	Allergen	
3	7 Hackbraten	a2	Roggen	Allergen	
4	9 Hühnersuppe	a2	Roggen	Allergen	
5	1 Bratkartoffeln mit Speck und Zwiebeln	a3	Gerste	Allergen	
6	4 Grilltofu	a3	Gerste	Allergen	
7	8 Gemüsepfanne	a3	Gerste	Allergen	
8	1 Bratkartoffeln mit Speck und Zwiebeln	a4	Dinkel	Allergen	
9	4 Grilltofu	a4	Dinkel	Allergen	
10	15 Pilzpfanne vegan	a4	Dinkel	Allergen	
11	12 Kassler mit Rosmarinkartoffeln	a5	Hafer	Allergen	
12	3 Bratkartoffeln mit Zwiebeln	a6	Dinkel	Allergen	
13	<null>	<null>	Fisch	Allergen	
14	7 Hackbraten	c	Krebstiere	Allergen	
15	1 Bratkartoffeln mit Speck und Zwiebeln	c	Krebstiere	Allergen	
16	6 Lasagne vegetarisch	d	Schwefeldioxid/Sulfit	Allergen	
17	10 Forellenfilet	d	Schwefeldioxid/Sulfit	Allergen	
18	<null>	<null>	Sellerie	Allergen	
19	10 Forellenfilet	f	Milch und Laktose	Allergen	
20	3 Bratkartoffeln mit Zwiebeln	f1	Butter	Allergen	
21	4 Grilltofu	f1	Butter	Allergen	
22	1 Bratkartoffeln mit Speck und Zwiebeln	f1	Butter	Allergen	

4) Die Anzahl der Gerichte pro Kategorie aufsteigend sortiert nach Anzahl.

```
SELECT k.*,COUNT(ger.id) as Anzahl FROM kategorie k LEFT JOIN
gericht_hat_kategorie ghk on k.id = ghk.kategorie_id LEFT JOIN gericht ger
on ghk.gericht_id = ger.id GROUP BY k.id ORDER BY Anzahl ASC
```

	id	name	eltern_id	bildname	Anzahl
1	6	Mensastars	1	kat_stars.tif	0
2	id: bigint	ktionen	<null>	kat_aktionen.png	0
3	7	Erstiewoche	1	kat_erties.jpg	0
4	2	Menus	<null>	kat_menu.gif	0
5	4	Vorspeisen	2	kat_menu_vor.svg	3
6	5	Desserts	2	kat_menu_dessert.pic	3
7	3	Hauptspeisen	2	kat_menu_haupt.bmp	7

5) Ändern Sie die vorherige Abfrage so ab, dass dabei nur die Kategorien dargestellt werden, die mehr als 2 Gerichte besitzen.

```
SELECT k.*,COUNT(ger.id) as Anzahl FROM kategorie k LEFT JOIN
gericht_hat_kategorie ghk on k.id = ghk.kategorie_id LEFT JOIN gericht ger
on ghk.gericht_id = ger.id GROUP BY k.id HAVING Anzahl > 2 ORDER BY Anzahl
ASC
```

	id	name	eltern_id	bildname	Anzahl
1	4	Vorspeisen		kat_menu_vor.svg	3
2	5	Desserts		kat_menu_dessert.pic	3
3	3	Hauptspeisen		kat_menu_haupt.bmp	7

6) (optional) Alle Gerichte, die vier oder mehr Allergene aufweisen.

```
SELECT * FROM gericht ger JOIN gericht_hat_allergen g ON ger.id =
g.gericht_id JOIN allergen al on g.code = al.code GROUP BY id HAVING
COUNT(al.code) >= 4
```

	id	name	Anzahl_Allergen
1	1	Bratkartoffeln mit Speck und Zwiebeln	5
2	4	Grilltofu	4
3	7	Hackbraten	4
4	15	Pilzpfanne vegan	5

Aufgabe 7

```
alter table gericht_hat_kategorie
add constraint gericht_hat_kategorie_fk
foreign key (`kategorie_id`) references kategorie (`id`);
```

```
alter table gericht_hat_kategorie
add constraint gericht_hat_kategorie_fk
foreign key (`gericht_id`) references gericht (`id`);
```

```
alter table gericht_hat_allergen
add constraint gericht_hat_allergen_fk
foreign key (`gericht_id`) references gericht (`id`);
```

```
alter table gericht_hat_allergen
add constraint gericht_hat_allergen_fk
foreign key (`code`) references allergen (`code`);
```

Aufgabe 11

- 1) Warum kann das Speichern in einfachen Dateien (ohne Einsatz von Datenbanken) direkt aus der Anwendung heraus Probleme bereiten?
 - Mit einfachen Dateien wird es beim Suchen mehr komplizierter als Datenbanken z.B wenn es sehr viele Gerichte schon in E-Mensa Werbeseite gab und wir nach einem spezifischen Preis suchen wollen, müssen wir zuerst jede Datei einzeln checken und nicht nur unpraktisch sondern auch viel Zeit verbraucht wird.
 - Mit Datenbanken können wir ziemlich großen Speicherplatz sparen, weil Datenbanken in Tabellen dargestellt werden.

2) b) & c)

Kontakt(Kontaktid, Postleitzahl, Straße, Ort, Hausnummer, Vorname, Nachname, Telefonnummer)

Kunde(Kontaktid, Postleitzahl, Straße, Ort, Hausnummer, Vorname, Nachname, Telefonnummer)

Zulieferer(Kontaktid, Postleitzahl, Straße, Ort, Hausnummer, Vorname, Nachname, Telefonnummer)

Anrede(Kontaktid)

Herr(Kontaktid)

Frau(Kontaktid)

Lieferungszeitfenster(Kontaktid)

morgens(Kontaktid)

mittags(Kontaktid)

abends(Kontaktid)

Hausanschrift(Kontaktid, Straße, Hausnummer, Postleitzahl, Ort)

Hauptanschrift(Kontaktid, Straße, Hausnummer, Postleitzahl, Ort)

Zweiwohnsitz(Kontaktid, Straße, Hausnummer, Postleitzahl, Ort)

Rückfragen(Kontaktid, Rückfragenummer, Text, Erfassungszeitpunkt, Dringlichkeit, Alter)

Thema(Rückfragenummer)

Beschwerde(Rückfragenummer)

Frage(Rückfragenummer)

Wunsch(Rückfragenummer)

***Kursiv und Unterstreichen = Fremdschlüssel (durch gestrichelte Unterstreichung)**

Meilenstein	Aufgabe	Geschätzte Zeit (in Minuten)	Tatsächlich benötigte Zeit (in Minuten)
4	1	75	120
	2	60	40
	3	-	-
	4	60	75
	uns wurden die Queries noch nicht bekannt		
	5	30	30
	6	10	10
	7	60	60
	8	75	75

Aufgabe 1.1

Wunschgericht(Name,E-mail,Gericht,Beschreibung,date,id)

Aufgabe 2C

SQL Injection

- Werbeseite

vorher

```
$Name = trim($_POST['Name'] ?? NULL);
$sprache=$_POST['sprache'];
$email = trim($_POST['E-Mail'] ?? NULL);
```

jetzt

```
$Name = str_replace("\'", "'", trim($_POST['Name'] ?? NULL));
$sprache=$_POST['sprache'];
$email = str_replace("\'", "'", trim($_POST['E-Mail'] ?? NULL));
```

vorher

```
'Name'=>$Name,
'E-Mail'=>$Email,
'Sprache'=>$sprache,
```

jetzt

```
'Name'=>mysqli_real_escape_string($link,$Name),
'E-Mail'=>mysqli_real_escape_string($link,$Email),
'Sprache'=>$sprache,
```

- Wunschgericht

vorher

```

$Name=trim($_POST['Name']??NULL);
$email =trim($_POST['E-Mail']??NULL));
$Gericht=trim($_POST['Namegericht']??NULL);
$Beschreibung=trim($_POST['Beschreibung']??NULL);

```

jetzt

```

$Name=str_replace("\'", "'", trim($_POST['Name']??NULL));
$email =str_replace("\'", "'", trim($_POST['E-Mail']??NULL));
$Gericht=str_replace("\'", "'", trim($_POST['Namegericht']??NULL));
$Beschreibung=str_replace("\'", "'", trim($_POST['Beschreibung']??NULL));

```

vorher

```

$input = ['Name'=>$Name,
          'E-Mail'=>$Email,
          'Gericht'=>$Gericht,
          'Beschreibung'=>$Beschreibung
];

```

jetzt

```

$input = ['Name'=>mysqli_real_escape_string($link,$Name),
          'E-Mail'=>mysqli_real_escape_string($link,$Email),
          'Gericht'=>mysqli_real_escape_string($link,$Gericht),
          'Beschreibung'=>mysqli_real_escape_string($link,$Beschreibung)
];

```

Aufgabe 4

- 1)

```
alter table gericht_hat_kategorie
add constraint unique_id unique (gericht_id,kategorie_id);
```
- 2)

```
create index Name_indexiert on gericht(name);
```
- 3)

```
alter table gericht_hat_kategorie
drop foreign key gericht_hat_kategorie_gericht_null_fk;

alter table gericht_hat_kategorie
add constraint gericht_hat_kategorie_gericht_null_fk
foreign key (gericht_id) references gericht (id)
on delete cascade;

alter table gericht_hat_allergen
drop foreign key gericht_hat_allergen_gericht_fk;

alter table gericht_hat_allergen
add constraint gericht_hat_allergen_gericht_fk
foreign key (gericht_id) references gericht (id)
on delete cascade;
```
- 4)

```
alter table kategorie
drop foreign key kategorie_kategorie_fk;
```

```
alter table kategorie
  add constraint kategorie_kategorie_fk
    foreign key (eltern_id) references kategorie (id)
    on delete restrict ;
```

```
alter table gericht_hat_kategorie
  drop foreign key gericht_hat_kategorie_kategorie_null_fk;
```

```
alter table gericht_hat_kategorie
  add constraint gericht_hat_kategorie_kategorie_null_fk
    foreign key (kategorie_id) references kategorie (id)
    on delete restrict ;
```

5)

```
alter table gericht_hat_allergen
  drop foreign key gericht_hat_allergen_allergen_code_fk;
```

```
alter table gericht_hat_allergen
  add constraint gericht_hat_allergen_allergen_code_fk
    foreign key (code) references allergen (code)
    on update cascade;
```

6)

```
alter table gericht_hat_kategorie
  add constraint pkey_comb primary key (gericht_id,kategorie_id);
```