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 bissc die Objekte und Ele	hen verwirrend, um mente 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.
- 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

- 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

Quele:

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 Bearbei von den Get-Parametern verwird	
	4	4	4
	5	120	240
		Wir haben PHP noch nicht gewöhn 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 minimalle 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/ParameterFloat : Datentyp für Dezimalzahlen

\$var : Variable
\$Var =[] : Array

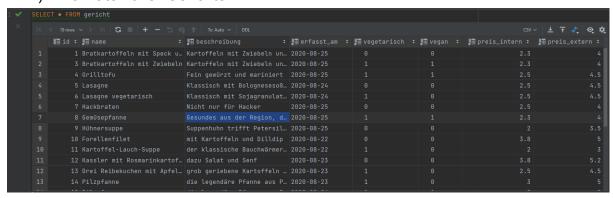
foreach : Array iterator if/else : Verzweigung. isset : Variable existiert

empty : Variablenwert nicht gleich Null

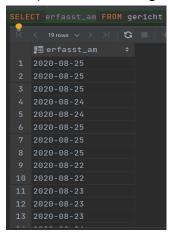
Meilenstein	Aufgabe	Geschätzte Zeit (in Minuten)	Tatsächlich benötigte Zeit (in Minuten)
3	1	10	8
	2	30	20
	3	45	40
	4	4	4
	5	60	45
	6	120	120
	7	45	45
	8	180	151
	9	120	110
	10	10	10
	11	90	90

Aufgabe 3

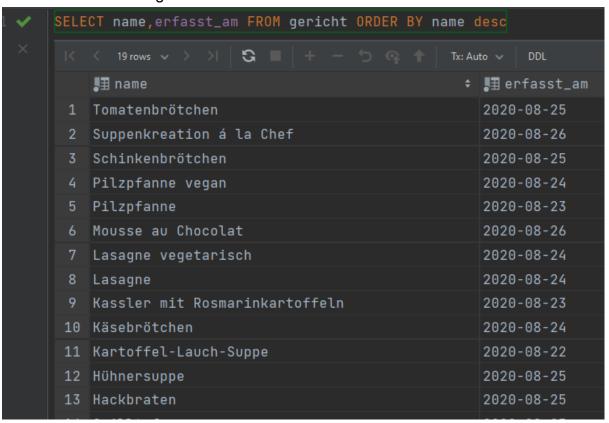
1) Alle Daten aller Gerichte



2) Das Erfassungsdatum aller Gerichte



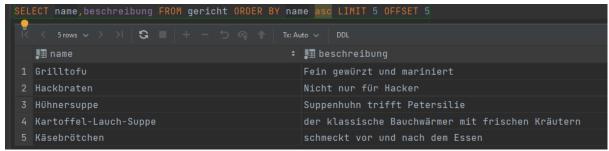
3) Das Erfassungsdatum sowie den Namen (als Attributname Gerichtname) aller Gerichte absteigend sortiert nach Gerichtname.



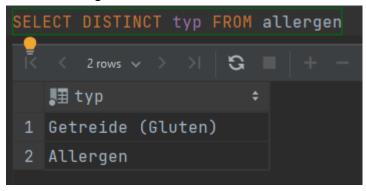
4) Den Namen sowie die Beschreibung der Gerichte aufsteigend sortiert nach Name, wobei nur 5 Datensätze dargestellt werden sollen.



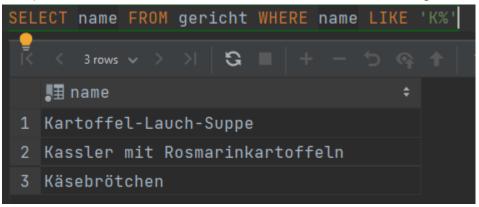
5) Ändern Sie die vorherige Abfrage so ab, so dass 10 Datensätze dargestellt werden, die nach den ersten 5 Datensätzen folgen. (Die ersten 5 Datensätze werden übersprungen)



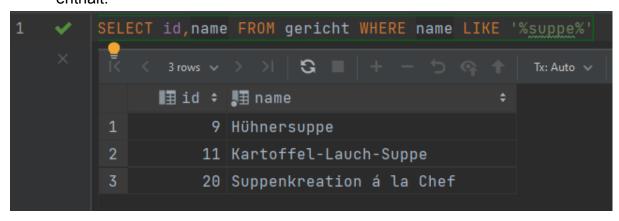
6) Zeigen Sie alle möglichen Allergen-Typen (typ), wobei Sie keine doppelten Einträge darstellen.



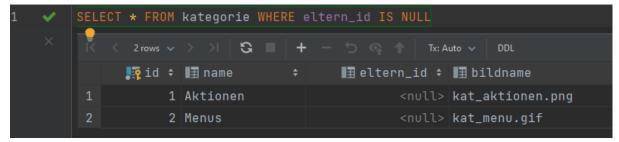
7) Namen von Gerichten, deren Name mit einem "K" beginnt.



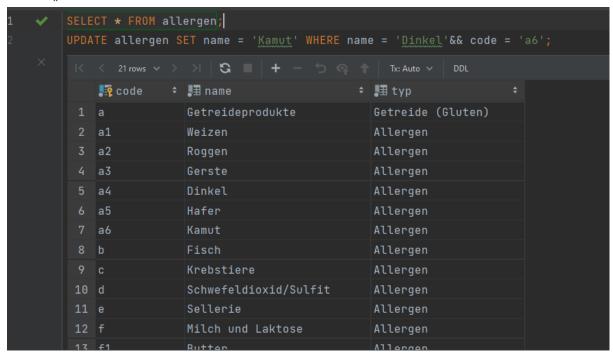
8) Ids und Namen von Gerichten, deren Namen ein "suppe" an beliebiger Stelle enthält.



9) Alle Kategorien, die keine Elterneinträge besitzen.



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



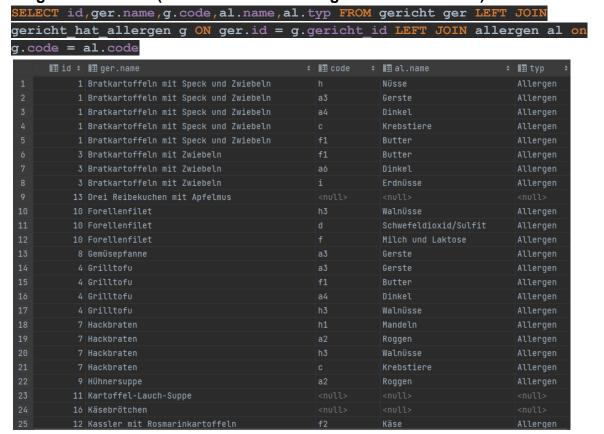
11) Fügen Sie das Gericht "Currywurst mit Pommes" hinzu und tragen Sie es in der Kategorie "Hauptspeise" (3) ein.



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 Ⅲ id ÷ Ⅲ ger.name ‡ **I** al.name 1 Bratkartoffeln mit Speck und Zwiebeln 1 Bratkartoffeln mit Speck und Zwiebeln Allergen 1 Bratkartoffeln mit Speck und Zwiebeln Allergen 1 Bratkartoffeln mit Speck und Zwiebeln Allergen 3 Bratkartoffeln mit Zwiebeln 3 Bratkartoffeln mit Zwiebeln 3 Bratkartoffeln mit Zwiebeln Allergen 10 Forellenfilet Milch und Laktose Allergen Allergen 4 Grilltofu Allergen Allergen Allergen 7 Hackbraten Allergen 7 Hackbraten Allergen 12 Kassler mit Rosmarinkartoffeln

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



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				Getreideprodukte	Getreide (Gluten)
2	15	Pilzpfanne vegan	a1	Weizen	Allergen
3		Hackbraten	a2	Roggen	Allergen
4		Hühnersuppe	a2	Roggen	Allergen
5		Bratkartoffeln mit Speck und Zwiebeln	а3	Gerste	Allergen
6		Grilltofu	а3	Gerste	Allergen
7		Gemüsepfanne	а3	Gerste	Allergen
8		Bratkartoffeln mit Speck und Zwiebeln	a4	Dinkel	Allergen
9		Grilltofu	a4	Dinkel	Allergen
10	15	Pilzpfanne vegan	a4	Dinkel	Allergen
11	12	Kassler mit Rosmarinkartoffeln	a5	Hafer	Allergen
12		Bratkartoffeln mit Zwiebeln	aó	Dinkel	Allergen
13				Fisch	Allergen
14		Hackbraten		Krebstiere	Allergen
15		Bratkartoffeln mit Speck und Zwiebeln		Krebstiere	Allergen
16		Lasagne vegetarisch		Schwefeldioxid/Sulfit	Allergen
17	10	Forellenfilet		Schwefeldioxid/Sulfit	Allergen
18				Sellerie	Allergen
19	10	Forellenfilet		Milch und Laktose	Allergen
20		Bratkartoffeln mit Zwiebeln	f1	Butter	Allergen
21		Grilltofu	f1	Butter	Allergen
22		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	: I⊞ eltern_id :	聞 bildname ÷	∎ Anzahl ‡
1	6	Mensastars	1	kat_stars.tif	Θ
2	id: bigir	^{nt} tionen	<null></null>	kat_aktionen.png	Θ
3	7	Erstiewoche	1	kat_erties.jpg	Θ
4	2	Menus	<null></null>	kat_menu.gif	Θ
5		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 :	Ⅲ bildname	I ≣ Anzahl ‡
1		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 ÷	III name ÷	■ Anzahl_Allergen ÷
1	1	Bratkartoffeln mit Speck und Zwiebeln	5
2		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
   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(<u>Kontaktid</u>,Postleitzahl,Straße,Ort,Hausnummer,Vorname,Nachname, Telefonnummer)

Kunde(<u>Kontaktid</u>,Postleitzahl,Straße,Ort,Hausnummer,Vorname,Nachname, Telefonnummer)

Zulieferer(<u>Kontaktid</u>,Postleitzahl,Straße,Ort,Hausnummer,Vorname, Nachname,Telefonnummer)

Anrede(*Kontaktid*) Herr(*Kontaktid*) Frau(*Kontaktid*)

Lieferungszeitfenster(<u>Kontaktid</u>) morgens(<u>Kontaktid</u>) mittags(<u>Kontaktid</u>) abends(<u>Kontaktid</u>)

Hausanschrift(<u>Kontaktid</u>,Straße, Hausnummer, Postleitzahl, Ort) Hauptanschrift(<u>Kontaktid</u>,Straße, Hausnummer, Postleitzahl, Ort) Zweiwohnsitz(<u>Kontaktid</u>,Straße, Hausnummer, Postleitzahl, Ort)

Rückfragen(<u>Kontaktid</u>,<u>Rückfragenummer</u>,Text,Erfassungszeitpunkt, Dringlichkeit,Alter)

Thema(<u>Rückfragenummer</u>)
Beschwerde(<u>Rückfragenummer</u>)
Frage(<u>Rückfragenummer</u>)
Wunsch(<u>Rückfragenummer</u>)

*Kursiv und Unterstreichen = Fremdschlüssel (durch gestrichelte Unterstreichung)

Meilenstein	Aufgabe	Geschätzte Zeit (in Minuten)	Tatsächlich benötigte Zeit (in Minuten)
4	1	75	65
	2	60	40
	3	-	-
	4	60	75
	uns wu	cht 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);
      $Gericht=trim($ POST['Namegericht']??NULL);
      $Beschreibung=trim($ POST['Beschreibung']??NULL);
      ietzt
      $Name=str replace("\'","",trim($ 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,
        'Gericht'=>$Gericht,
      jetzt
      $input =['Name'=>mysqli_real_escape_string($link,$Name),
         'E-Mail'=>mysqli_real_escape_string($link,$Email),
         'Beschreibung'=>mysqli real escape string($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;
       lter 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;
         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)
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);
```

Meilenstein	Aufgabe	Geschätzte Zeit (in Minuten)	Tatsächlich benötigte Zeit (in Minuten)
5	1	180	115
	2	60	40
	3	60	60
	4	30	20
	5	30	30
Total			265

Aufgabe 5C

- 1) Datum für letzteanmeldungen und letztefehler
- 2) anzahlfehler bei Anmeldungfehler inkrementiert

Meilenstein	Aufgabe	Geschätzte Zeit (in Minuten)	Tatsächlich benötigte Zeit (in Minuten)
6	1	300	285
	2	180	135
	3	-	-
Total		450	420