**Dokumentation**

## Einstieg:

Moin,

wir haben unsere Datenbank einmal als .sql Export beigefügt, damit ihr die Abfragen auch Lokal direkt testen könnt, bestenfalls solltet ihr die Datei nur starten müssen und könnt dann aus dem Abfragen-Ordner die Abfragen ausprobieren.

**Möglichkeit 2** wir haben auch eine kleine Webseite geschrieben unter der ihr die Abfragen testen könnt, das ganze findet ihr unter [bs14.code-daniel.tech](http://www.bs14.code-daniel.tech/). Wichtig dazu, das ganze ist aktuell nur Desktop kompatibel und auch nicht wirklich responsive, irgendwann wollte ich dann doch nochmal schlafen. Ansonsten sollte das so laufen, bei Fragen zum UI einfach mal auf „Gemüse bestellen“ gehen.

## Beispiele, die ihr für das Web-UI verwenden könnt

(das Web-UI findet ihr unter [bs14.code-daniel.tech](http://www.bs14.code-daniel.tech/))

#### **Select Rezept bei Rezeptname**

* Thaicurry mit Hähnchen
* Lachslasagne
* Kartoffelsuppe

#### **Select Rezept bei Zutat**

* Lachs
* Butter
* Reis
* Kartoffel

#### **Select Preis bei Rezeptnummer**

* 5
* 6
* 7

#### **Berechnet Preis für Bestellung**

* 13
* 14

#### **Auswahl aller Rezepte die Kalorienlimit unterschreiten**

* 3500
* 42069

(nur als Hilfe, erwartet halt bei allem anderen kein Output)

## Tabelle mit allen Abfragen für unsere Datenbank

|  |  |  |
| --- | --- | --- |
| Abfrage | Beschreibung | Check |
| select\_nach\_rezeptname.sql | Auswahl eines Rezeptes nach Rezeptname |  |
| select\_rezept\_nach\_zutat.sql | Auswahl aller Rezepte, die eine bestimmte Zutat enthalten |  |
| avg\_nährwert\_von\_kunden.sql | Berechnung der durchschnittlichen Nährwerte aller Bestellungen eines Kunden |  |
| select\_zutat\_ohne\_rezept.sql | Auswahl aller Zutaten, die in keime Rezept vorkommen |  |
| select\_kunde\_eine\_bestellung.sql | Auswahl aller Kunden, die nur einmal bestellt haben |  |
| berechnung\_bestell\_preis\_rezept.sql | Berechnung des Preises der gesamten Bestellung |  |
| berechnung\_rezept\_preis.sql | Berechnung des Preises für ein Rezept |  |
| select\_rezept\_weniger\_als\_5\_zutaten.sql | Auswahl aller Rezepte, die weniger als 5 Zutaten enthalten |  |
| select\_rezept\_weniger\_als\_5\_zutaten\_  [und\_Ernährungskategorie](file:///C:\KrautundRüben\Abfragen\select_rezept_weniger_als_5_zutaten_und_ernahrungskategorie.sql).sql | Auswahl aller Rezepte, die weniger als 5 Zutaten enthalten und eine bestimmte Ernährungskategorie erfüllen |  |
| select\_rezept\_unter\_bestimmter\_\_kalorienmenge.sql | Auswahl aller Rezepte unter Kalorienangabe XY |  |

## Unser Sicherheitskonzept

**Starke Passwörter** geben im Fall eines Brute-Force-Angriffs (Das wahllose automatisiertes ausprobieren von verschieden Zeichenkombination, die ggf. das Passwort ergeben) mehr Zeit zum Handeln. Abhängig von der Länge und Komplexität der Passwörter kann es dann Jahre dauern bis diese herausgefunden werden können.

**Notwendigkeit der Benutzer und Zugriffskontrolle** gehen miteinander einher. Braucht ein Backend das Datenausließt und wiedergeben soll, Berechtigungen auf Adminebene? Unwahrscheinlich, genauso bringen unterschiedliche Accounts für Teammitglieder, mehr nach Vollziehbarkeit im Schadensfall.

**Sonstige** andere Möglichkeiten der Sicherheit bspw. SSL oder 2FA existieren und sind auch umsetzbar, finden bei uns aber keine Anwendung aus Zeitlichen Gründen.

Unsere Datenschutzerklärung ist zu finden unter [bs14.code-daniel.tech/datenschutz.html.](https://www.bs14.code-daniel.tech/datenschutz.html)

## Unser ERD-Modell

