**Handout für das Projekt "Rezept- und Zutatendatenbank"**

**Projektname:** Rezept- und Zutatendatenbank

**Abgabedatum:** [Bitte hier das Abgabedatum einfügen]

**1. Einleitung**

In diesem Projekt haben wir eine relationale Datenbank entwickelt, die eine Vielzahl von Rezepten und Zutaten verwaltet und es ermöglicht, Rezepte basierend auf verschiedenen Kriterien zu suchen und zu filtern. Die Datenbank enthält auch die Verwaltung von Nährwerten, Allergenen und CO2-Äquivalenten, die für jedes Lebensmittel erfasst werden. Dies bietet eine fundierte Grundlage für die Erstellung von Ernährungsplänen, Analyse von Bestellungen und die Integration von Ernährungsrichtlinien.

**2. Ziel des Projekts**

Das Ziel dieses Projekts war es, eine voll funktionsfähige Datenbank zu entwickeln, die:

* **Rezepte speichert**, die aus mehreren Zutaten bestehen.
* **Ernährungskategorien** für jedes Rezept verwaltet.
* **Allergene** für jedes Rezept speichert und berücksichtigt.
* **Bestellungen von Kunden** inklusive Nährwertberechnungen ermöglicht.
* **CO2-Äquivalente** für Zutaten hinzufügt.
* Einfache **Abfragen** ermöglicht, um Rezepte nach verschiedenen Kriterien wie Zutaten, Kalorienanzahl, Nährwerten und mehr zu suchen.

**3. Datenbankstruktur und Normalisierung**

Die Datenbank wurde in der **dritten Normalform (3NF)** entworfen, um Redundanzen zu vermeiden und Datenkonsistenz zu gewährleisten. Die Struktur umfasst die folgenden wesentlichen Tabellen:

* **Kunde:** Speichert die Daten der Kunden, die Bestellungen aufgeben.
* **Zutat:** Enthält alle Zutaten, ihre Nährwerte und CO2-Äquivalente.
* **Rezept:** Speichert Informationen zu Rezepten, inklusive Zubereitung und Kategorie.
* **Bestellung:** Speichert Bestellungen von Kunden, die Zutatendaten enthalten.
* **Allergen:** Enthält Informationen zu möglichen Allergenen.
* **Ernaehrungskategorie:** Verwaltet verschiedene Ernährungskategorien wie "Vegan", "Low Carb", "Glutenfrei".
* **Rezept\_zutat:** Verknüpft Rezepte und Zutaten.

**4. SQL-Abfragen und Anforderungen**

Die wichtigsten SQL-Abfragen, die in der Datenbank verwendet werden, sind wie folgt:

**Beispielhafte SQL-Abfragen:**

1. **Auswahl aller Zutaten eines Rezeptes nach Rezeptname:**

sql

KopierenBearbeiten

SELECT z.zutat\_name, rz.menge

FROM Rezept\_zutat rz

JOIN Zutat z ON rz.zutat\_id = z.zutat\_id

JOIN Rezept r ON rz.rezept\_id = r.rezept\_id

WHERE r.rezept\_name = 'Gemüsepfanne';

1. **Auswahl aller Rezepte einer bestimmten Ernährungskategorie:**

sql

KopierenBearbeiten

SELECT r.rezept\_name

FROM Rezept r

JOIN Ernaehrungskategorie ek ON r.kategorie\_id = ek.kategorie\_id

WHERE ek.kategorie\_name = 'Vegan';

1. **Berechnung der durchschnittlichen Nährwerte aller Bestellungen eines Kunden:**

sql

KopierenBearbeiten

SELECT

kd.kd\_nr,

AVG(n.fett\_pro\_einheit) AS avg\_fett,

AVG(n.kalorien\_pro\_einheit) AS avg\_kalorien,

AVG(n.kohlenhydrat\_pro\_einheit) AS avg\_kohlenhydrate,

AVG(n.protein) AS avg\_protein

FROM Bestellung b

JOIN Kunde kd ON b.kd\_nr = kd.kd\_nr

JOIN Bestellung\_zutat bz ON b.bestell\_nr = bz.bestell\_nr

JOIN Zutat z ON bz.zutat\_id = z.zutat\_id

JOIN Naehrwert n ON z.naehrwert\_id = n.naehrwert\_id

GROUP BY kd.kd\_nr;

1. **Auswahl aller Rezepte, die eine bestimmte Kalorienmenge nicht überschreiten:**

sql

KopierenBearbeiten

SELECT r.rezept\_name

FROM Rezept r

JOIN Rezept\_zutat rz ON r.rezept\_id = rz.rezept\_id

JOIN Zutat z ON rz.zutat\_id = z.zutat\_id

JOIN Naehrwert n ON z.naehrwert\_id = n.naehrwert\_id

GROUP BY r.rezept\_name

HAVING SUM(n.kalorien\_pro\_einheit) <= 500; -- Beispiel für 500 Kalorien

**5. Besondere Features und Anforderungen**

* **CO2-Äquivalente:** Jede Zutat hat einen CO2-Wert, der die Umweltbelastung durch die Produktion angibt. Dies wurde als zusätzlicher Wert in der Zutatentabelle hinzugefügt.
* **Allergene:** Die Zutatentabelle ist mit einer weiteren Tabelle Allergen verknüpft, in der spezifische Allergene wie Gluten, Nüsse oder Laktose aufgeführt sind. In der Tabelle Rezept\_allergen wird jedes Rezept mit den enthaltenen Allergenen verknüpft.
* **Ernährungskategorien:** Die Rezepte werden in verschiedene Ernährungskategorien eingeteilt (z. B. "Vegan", "Glutenfrei", "Low Carb"). Dies hilft, Rezepte schnell nach speziellen Ernährungsbedürfnissen zu filtern.
* **Bestellungen:** Kundenbestellungen werden aufgezeichnet und mit Zutaten verknüpft. Die Nährwerte der Bestellungen können mit den entsprechenden SQL-Abfragen berechnet werden, um die durchschnittlichen Nährwerte pro Kunde zu ermitteln.

**6. DSGVO-konforme Implementierungen**

Im Rahmen des Projekts wurden auch zwei grundlegende Funktionen zur Einhaltung der **Datenschutz-Grundverordnung (DSGVO)** implementiert:

* **Auskunftsrecht:** Ein Kunde hat das Recht, Auskunft über die von ihm gespeicherten Daten zu erhalten. Die entsprechenden SQL-Abfragen bieten eine Möglichkeit, alle gespeicherten Bestell- und Kundendaten abzurufen.

sql

KopierenBearbeiten

SELECT \* FROM Kunde WHERE kd\_nr = ?;

* **Recht auf Löschung:** Ein Kunde kann die Löschung seiner Daten verlangen. Die Löschfunktion entfernt sowohl die Kundeninformationen als auch alle damit verbundenen Bestell- und Zutatendaten:

sql

KopierenBearbeiten

DELETE FROM Kunde WHERE kd\_nr = ?;

DELETE FROM Bestellung WHERE kd\_nr = ?;

DELETE FROM Bestellung\_zutat WHERE bestell\_nr IN (SELECT bestell\_nr FROM Bestellung WHERE kd\_nr = ?);

**7. Weiterentwicklungsmöglichkeiten**

In Zukunft könnte die Datenbank um folgende Funktionen erweitert werden:

* **Rezeptbewertungen:** Kunden könnten Rezepte bewerten und Kommentare hinterlassen.
* **Personalisierte Ernährung:** Kunden könnten Ernährungspräferenzen festlegen, und die Datenbank könnte automatisch Rezepte empfehlen.
* **Integration von externen Nährwertdaten:** Eine API könnte verwendet werden, um Nährwerte und CO2-Werte für eine größere Anzahl von Lebensmitteln zu integrieren.

**8. Schlussfolgerung**

Das Projekt "Rezept- und Zutatendatenbank" erfüllt alle vorgegebenen Anforderungen und stellt sicher, dass die Rezepte, Zutaten, Bestellungen und Nährwertdaten korrekt und effizient verwaltet werden. Die Datenbank ist für die Nutzung in realen Szenarien gut geeignet und bietet eine solide Grundlage für die Integration zusätzlicher Funktionen in der Zukunft.