

# Kraut und Rüben SQL-Dokumentation

Zeige alle Zutaten, die in einem Rezept sind (1):

```
SELECT ZUTAT.ZUTATENNR, ZUTAT.BEZEICHNUNG, REZEPTZUTATEN.MENGE,
REZEPTZUTATEN.EINHEIT
FROM( ( ZUTAT

INNER JOIN REZEPTZUTATEN ON ZUTAT.ZUTATENNR = REZEPTZUTATEN.ZUTATENNR )

INNER JOIN REZEPTE ON REZEPTZUTATEN.REZEPTNR = REZEPTE.REZEPTNR )

WHERE REZEPT = "Kartoffelsuppe";
```

Erklärung:

Im String muss der Rezeptname angegeben werden. Um ein Vertippen zu vermeiden, kann man auch den Operator **LIKE** verwenden. Die Syntax sieht dann wie folgt aus:

```
WHERE REZEPT LIKE "%kart%";
```

Durch die Prozentzeichen findet man alle Werte, die an irgendeiner Stelle "kart" enthalten.

---

Zeige Rezepte in Ernährungskategorie (2):

```
SELECT REZEPTKATEGORIEN.REZEPTNR, REZEPTE.REZEPT,
REZEPTKATEGORIEN.KATEGORIENR, ERNAHRUNGSKATEGORIEN.KATEGORIE,
ERNAHRUNGSKATEGORIEN.BESCHREIBUNG
FROM( ( REZEPTE

LEFT JOIN REZEPTKATEGORIEN ON REZEPTE.REZEPTNR =
REZEPTKATEGORIEN.REZEPTNR )

RIGHT JOIN ERNAHRUNGSKATEGORIEN ON REZEPTKATEGORIEN.KATEGORIENR =
ERNAHRUNGSKATEGORIEN.KATEGORIENR )

WHERE KATEGORIE = "Vegan";
```

Erklärung:

Im String muss der Rezeptname angegeben werden. Um ein Vertippen zu vermeiden, kann man auch den Operator **LIKE** verwenden. Die Syntax sieht dann wie folgt aus:

```
WHERE KATEGORIE LIKE "%veg%";
```

Durch die Prozentzeichen findet man alle Werte, die an irgendeiner Stelle "veg" enthalten.

---

Rezepte nach Allergen ausschließen (3):

Erklärung:

```
SELECT REZEPTE.REZEPTNR, REZEPTE.REZEPT, ALLERGENE.ALLERGENNR,
ALLERGENE.ALLERGEN
FROM( ( REZEPTE

INNER JOIN REZEPTALERGENE ON REZEPTE.REZEPTNR = REZEPTALERGENE.REZEPTNR)
INNER JOIN ALLERGENE ON REZEPTALERGENE.ALLERGENNR = ALLERGENE.ALLERGENNR)

WHERE ALLERGEN != "gluten";
```

# Kraut und Rüben SQL-Dokumentation

Im String schreibt man das Allergen, dass man nicht in einem Rezept haben will. Wenn man mehrere Allergene ausschließen will, kann man zwischen den ersten String und dem Semikolon auch ein **AND ALLERGEN != "laktose"** hängen was dafür sorgt das beide Allergene von der Suche ausgeschlossen werden.

Zutat nach Allergen ausschließen (3.1):

Erklärung:

```
SELECT ZUTAT.ZUTATENNR, ZUTAT.BEZEICHNUNG, ALLERGENE.ALLERGENNR,
ALLERGENE.ALLERGEN
FROM( ( ZUTAT

INNER JOIN ZUTATENALERGENE ON ZUTAT.ZUTATENNR =
ZUTATENALERGENE.ZUTATENNR)
INNER JOIN ALLERGENE ON ZUTATENALERGENE.ALLERGENNR =
ALLERGENE.ALLERGENNR)

WHERE ALLERGEN != "gluten";
```

Im String schreibt man das Allergen, dass man nicht in einer Zutat haben will.

---

Rezept nach Namen auswählen und Beschreibung, Mahlzeit und Aufwand erhalten (4):

```
SELECT REZEPTE.REZEPTNR, REZEPTE.REZEPT, REZEPTE.ZUBEREITUNG,
REZEPTE.PORTIONEN, MAHLZEITE.BEZEICHNUNG AS MAHLZEIT,
AUFWANDE.BEZEICHNUNG AS AUFWAND
FROM( ( REZEPTE

LEFT JOIN MAHLZEITE ON REZEPTE.MAHLZEIT = MAHLZEITE.MAHLZEITNR)
LEFT JOIN AUFWANDE ON REZEPTE.AUFWAND = AUFWANDE.AUFWANDENR )

WHERE REZEPT = "Lachslasagne";
```

Erklärung:

Man kann anstatt den Rezeptnamen auch nach der Rezept Nummer suchen. Dies macht man, indem man **REZEPT** mit **REZEPTNR** ersetzt und den String durch eine Zahl (INT) ersetzt oder die Zahl in den String schreibt. Das verhindert das Verschreiben des Rezeptnamens. Man kann auch **INNER JOIN** verwenden.

---

```
SELECT REZEPTZUTATEN.REZEPTNR, REZEPTE.REZEPT, REZEPTZUTATEN.ZUTATENNR,
ZUTAT.BEZEICHNUNG, REZEPTZUTATEN.MENGE, REZEPTZUTATEN.EINHEIT
FROM( ( REZEPTE

INNER JOIN REZEPTZUTATEN ON REZEPTE.REZEPTNR = REZEPTZUTATEN.REZEPTNR )
INNER JOIN ZUTAT ON REZEPTZUTATEN.ZUTATENNR = ZUTAT.ZUTATENNR )

WHERE BEZEICHNUNG = "Reis";
```

Zeige alle Rezepte, die eine gewisse Zutat enthalten (5):

Erklärung:

# Kraut und Rüben SQL-Dokumentation

Mit der Bezeichnung ist der Name der Zutat gemeint.

```
SELECT KUNDE.KUNDENNR, KUNDE.NACHNAME,  
ROUND(AVG(ZUTAT.KALORIEN),1) AS "Ø Kalorien",  
ROUND(AVG(ZUTAT.KOHLLENHYDRATE),1) AS "Ø Kohlenhydrate",  
ROUND(AVG(ZUTAT.PROTEIN),1) AS "Ø Protein"  
  
FROM BESTELLUNG  
NATURAL JOIN BESTELLUNGSZUTAT  
NATURAL JOIN ZUTAT  
NATURAL JOIN KUNDE  
  
WHERE KUNDENNR = 2001;
```

Durchschnitts Nährstoffe Bestellung von einer gewissen Kundennummer (6):

Erklärung:

Die Kundennummer kann man als String sowie nur als Zahl schreiben da die Kundennummer ein INT ist.

---

```
SELECT ZUTAT.ZUTATENNR, ZUTAT.BEZEICHNUNG FROM ZUTAT  
  
LEFT JOIN REZEPTZUTATEN ON REZEPTZUTATEN.ZUTATENNR = ZUTAT.ZUTATENNR  
  
WHERE REZEPTZUTATEN.ZUTATENNR IS NULL;
```

Zeige Zutaten, die in keinem Rezept sind / keinem zugewiesen sind (7):

Erklärung:

Dieser Befehl gibt alle Zutaten zurück, die bei dem Left Join, den Wert **NULL** haben.

---

```
SELECT REZEPTE.REZEPTNR, REZEPTE.REZEPT, SUM(ZUTAT.KALORIEN) AS  
"Kalorien" FROM REZEPTE  
  
JOIN REZEPTZUTATEN ON REZEPTE.REZEPTNR = REZEPTZUTATEN.REZEPTNR  
JOIN ZUTAT ON ZUTAT.ZUTATENNR = REZEPTZUTATEN.ZUTATENNR  
GROUP BY REZEPTE.REZEPTNR  
  
HAVING SUM(ZUTAT.KALORIEN) < "1000";
```

Zeige Rezepte, die eine gewisse Kalorienanzahl nicht überschreiten (8):

Erklärung:

Die Kalorien kann man als String sowie nur als Zahl schreiben da die Kalorien ein INT ist.

# Kraut und Rüben SQL-Dokumentation

```
SELECT REZEPTRE.REZEPTNR, REZEPTRE.REZEPT, COUNT (REZEPTZUTATEN.ZUTATENNR)
AS "Zutatenanzahl" FROM REZEPTRE

JOIN REZEPTZUTATEN ON REZEPTRE.REZEPTNR = REZEPTZUTATEN.REZEPTNR
GROUP BY REZEPTZUTATEN.REZEPTNR

HAVING Zutatenanzahl <= 5;
```

Zeige alle Rezepte die 5 oder weniger Zutaten haben (9):

Erklärung:

Mit dem Operator `<=` ist es möglich Ergebnisse anzeigen zu lassen die den gleichen oder einen niedrigeren Wert haben als den INT danach.

---

Zeige Rezepte, die in einer gewissen Kategorie sind und 5 oder weniger Zutaten haben (10):

```
SELECT REZEPTRE.REZEPTNR, REZEPTRE.REZEPT, COUNT (REZEPTZUTATEN.ZUTATENNR)
AS "Zutatenanzahl", REZEPTKATEGORIEN.KATEGORIENR,
ERNAHRUNGSKATEGORIEN.KATEGORIE, ERNAHRUNGSKATEGORIEN.BESCHREIBUNG
FROM REZEPTRE
JOIN REZEPTZUTATEN ON REZEPTRE.REZEPTNR = REZEPTZUTATEN.REZEPTNR
JOIN REZEPTKATEGORIEN ON REZEPTRE.REZEPTNR = REZEPTKATEGORIEN.REZEPTNR
NATURAL JOIN ERNAHRUNGSKATEGORIEN

WHERE REZEPTKATEGORIEN.KATEGORIENR = "1"

GROUP BY REZEPTZUTATEN.REZEPTNR

HAVING COUNT (REZEPTZUTATEN.ZUTATENNR) <= 100 .
```

Erklärung:

Dieser Befehl hat zwei Eingabepunkte. Einmal. Die Kategorie Nummer die auch als reine Zahl geschrieben werden kann, da die Kategorie Nummer ein INT ist. Mit dem `HAVING COUNT` gibt man vor, nach wie vielen Zutaten gesucht werden, soll. Wie bei der Nummer 9. Mit dem `GROUP BY` sortiert man die Rezeptnummern nach Größe. In dem Fall wird die kleinste Nummer oben angezeigt und die größere weiter unten.

---

```
SELECT * FROM kunde
WHERE kunde.KUNDENNR = 2001;
```

Auskunft von Kundenbezogenen Daten nach DSGVO:

Erklärung:

Gibt den ganzen Eintrag aus, was mit dieser Kundennummer in dieser Tabelle existiert.

# Kraut und Rüben SQL-Dokumentation

---

```
DELETE FROM kunde
WHERE KUNDENNR = 2001;
```

Löschung von kundenbezogenen Daten nach DSGVO:

Erklärung:

Löscht den ganzen Eintrag, was mit dieser Kundennummer in dieser Tabelle existiert. Der Rest der Daten der nach der Löschung übrig bleibt sind die Kundennummer im Zusammenhang mit der Bestellnummer in der Tabelle **bestellung** und die Bestellnummer mit der Zutatennummer in der Tabelle **bestellungzutat**.

---

```
SELECT * FROM KUNDE
WHERE kunde.GEBURTSDATUM = (
    SELECT MIN(kunde.GEBURTSDATUM) FROM KUNDE
);
```

Zeige Kunden mit dem ältesten Geburtsdatum (Subselect):

Erklärung:

MySQL sucht den ältesten Kunden raus. Interessant für Personalisierung der Rezeptvorschläge angepasst auf das Alter.

---

```
SELECT REZEPTE.REZEPTNR, REZEPTE.REZEPT, REZEPTE.ZUBEREITUNG,
REZEPTE.PORCIONEN, AUFWANDE.BEZEICHNUNG AS AUFWAND
FROM( REZEPTE

LEFT JOIN AUFWANDE ON REZEPTE.AUFWAND = AUFWANDE.AUFWANDENR )
WHERE AUFWAND <= "2";
```

Zeige Rezepte die einen Aufwand von 2 (Mittel bis Gering) und geringer haben (Zusatz 1):

Erklärung:

Im String ist wird ein INT geschrieben. Dieser kann auch ohne Anführungsstriche geschrieben werden.

---

# Kraut und Rüben SQL-Dokumentation

```
SELECT REZEPTE.REZEPTNR, REZEPTE.REZEPT, REZEPTE.ZUBEREITUNG,
REZEPTE.PORTIONEN, MAHLZEITE.BEZEICHNUNG AS MAHLZEIT
FROM( REZEPTE

LEFT JOIN MAHLZEITE ON REZEPTE.MAHLZEIT = MAHLZEITE.MAHLZEITNR )

WHERE MAHLZEIT = 2;
```

Zeige Rezepte die eine Mahlzeit von 2 (Vorspeise) haben (Zusatz 2):

Erklärung:

Die INT kann man auch in einem String schreiben.

Zeige Rezepte die eine Mahlzeit von 2 (Vorspeise) und Aufwand von 2 (Mittel bis Gering) und geringer haben (Zusatz 3):

Erklärung:

```
SELECT REZEPTE.REZEPTNR, REZEPTE.REZEPT, REZEPTE.ZUBEREITUNG,
REZEPTE.PORTIONEN, MAHLZEITE.BEZEICHNUNG AS
MAHLZEIT,AUFWANDE.BEZEICHNUNG AS AUFWAND
FROM( ( REZEPTE

LEFT JOIN AUFWANDE ON REZEPTE.AUFWAND = AUFWANDE.AUFWANDENR )
LEFT JOIN MAHLZEITE ON REZEPTE.MAHLZEIT = MAHLZEITE.MAHLZEITNR )

WHERE AUFWAND <= "2" AND MAHLZEIT = "2";
```

Beide Strings kann man auch als INT, sprich ohne die Anführungsstriche schreiben.

---

```
SELECT ZUTAT.ZUTATENNR, ZUTAT.BEZEICHNUNG, ALLERGENE.ALLERGENNR,
ALLERGENE.ALLERGEN
FROM( ( ZUTAT

INNER JOIN ZUTATENALERGENE ON ZUTAT.ZUTATENNR =
ZUTATENALERGENE.ZUTATENNR)
INNER JOIN ALLERGENE ON ZUTATENALERGENE.ALLERGENNR =
ALLERGENE.ALLERGENNR)

WHERE ALLERGENE.ALLERGENNR = 100;
```

Zeige Zutaten, die kein Allergen haben (Zusatz 4):

Das Allergen mit der Nummer 100 hat die Bezeichnung „keine“.

---

# Kraut und Rüben SQL-Dokumentation