

Arbeitsauftrag A5.0 Beschreiben eines ER-Diagrammes

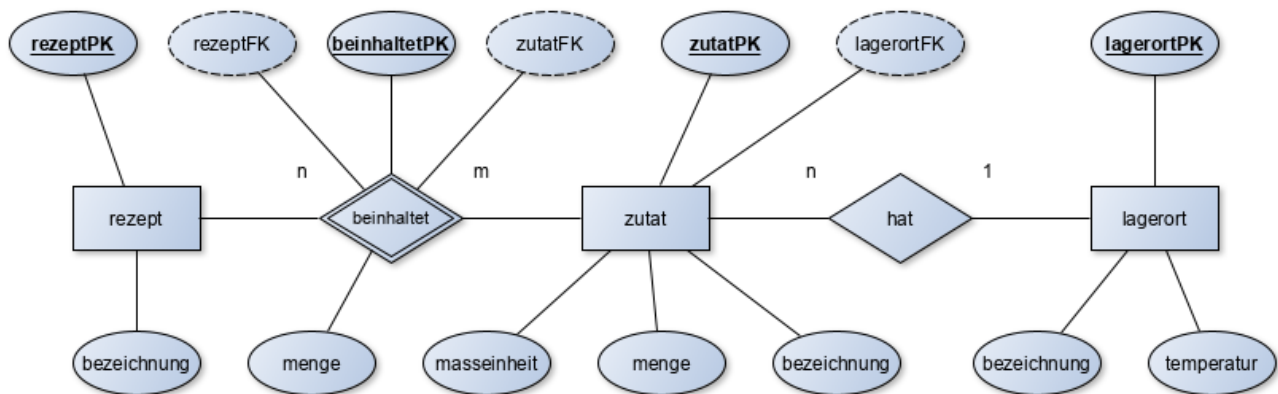
Ein Restaurant möchte eine bestehende Datenbank erweitern. Bisher sind einige Sachverhalte für die Vorratshaltung abgebildet:

- Ein Rezept hat mehrere Zutaten.
- Eine Zutat kann in mehreren Rezepten verwendet werden.
- Eine Zutat stammt immer genau aus einem Lagerort.
- Lagerorte werden für mehrere Zutaten genutzt.

So kann in dem Restaurant bisher folgendes dargestellt werden:

- Für den hauseigenen Vollkorn-Pizzateig werden Weizenmehl, Vollkornmehl, frische Hefe, Salz, Zucker, Olivenöl und Wasser benötigt.
- Vollkornmehl, Weizenmehl, Salz und Zucker werden im Trockenlager bevorratet.
- Frische Hefe muss in den Kühlschrank.
- Olivenöl ist im Flaschenlager zu finden.
- Eine Zutat hat eine Maßeinheit, bspw. wird Weizenmehl in Gramm (g) angegeben.
- Rezepte haben einen Namen, der auf der Karte ausgewiesen wird.
- ...

Beschreiben Sie mit eigenen Worten den Aufbau eines ER-Diagrammes immer in Bezug auf das Beispiel:



1. Was sind Entitäten?
2. Was sind Attribute?
3. Was sind Beziehungen?
4. Was sind Kardinalitäten?
5. Was bedeuten die im Beispiel eingesetzten Kardinalitäten in dem Fall?
6. Welche Aufgabe hat der Primärschlüssel?
7. Welche Aufgabe hat der Fremdschlüssel?
8. Was ist referenzielle Integrität?

Das ER-Diagramm am Beispiel des Rezepts

Entitäten:

- **Rezept:** Eine Entität repräsentiert ein einzelnes Rezept mit seinen Eigenschaften.
- **Zutat:** Eine Entität repräsentiert eine Zutat, die in einem Rezept verwendet wird, mit ihren Eigenschaften.
- **Lagerort:** Eine Entität repräsentiert einen Lagerort, an dem Zutaten aufbewahrt werden, mit seinen Eigenschaften.

Attribute:

- **Rezept:**
 - **rezeptPK** : Eindeutige Identifikationsnummer des Rezepts (Primärschlüssel)

- `bezeichnung` : Name des Rezepts
- **Zutat:**
 - `zutatPK` : Eindeutige Identifikationsnummer der Zutat (Primärschlüssel)
 - `bezeichnung` : Name der Zutat
 - `menge` : Menge der Zutat, die im Rezept verwendet wird
 - `masseinheit` : Masseinheit der Menge (z.B. Gramm, Stück)
- **Lagerort:**
 - `lagerortPK` : Eindeutige Identifikationsnummer des Lagerorts (Primärschlüssel)
 - `bezeichnung` : Name des Lagerorts
 - `temperatur` : Lagertemperatur für die Zutat (z.B. Kühlschrank, Raumtemperatur)

Beziehungen:

- **beinhaltet:** Ein Rezept beinhaltet mehrere Zutaten.
- **hat:** Eine Zutat kann in mehreren Rezepten verwendet werden.
- **lagert:** Eine Zutat wird in einem Lagerort aufbewahrt.

Kardinalitäten:

- **1:n:** Ein Rezept beinhaltet **eine oder mehrere** Zutaten (1:n). Das bedeutet, dass ein Rezept mindestens eine Zutat enthalten muss, aber auch mehrere Zutaten haben kann.
- **n:m:** Eine Zutat kann in **mehreren Rezepten** verwendet werden (n:m). Das bedeutet, dass eine Zutat in einem Rezept verwendet werden kann, aber auch in anderen Rezepten verwendet werden kann.
- **1:1:** Eine Zutat wird in **genau einem Lagerort** gelagert (1:1). Das bedeutet, dass eine Zutat nur in einem Lagerort aufbewahrt werden kann und ein Lagerort nur eine Zutat enthalten kann.

Im Beispiel:

- Ein Rezept kann mehrere Zutaten enthalten, z.B. Mehl, Zucker und Eier.
- Eine Zutat kann in mehreren Rezepten verwendet werden, z.B. Mehl kann in Kuchen, Brot und Pfannkuchen verwendet werden.
- Eine Zutat wird in einem Lagerort aufbewahrt, z.B. Mehl in der Speisekammer.

Primärschlüssel:

Der Primärschlüssel ist eine eindeutige Identifikationsnummer für jede Entität. In diesem Beispiel:

- `rezeptPK` für die Entität `Rezept`
- `zutatPK` für die Entität `Zutat`
- `lagerortPK` für die Entität `Lagerort`

Der Primärschlüssel dient dazu, Datensätze eindeutig zu identifizieren und sicherzustellen, dass es keine doppelten Datensätze in der Datenbank gibt.

Fremdschlüssel:

Ein Fremdschlüssel ist eine Spalte in einer Tabelle, die auf den Primärschlüssel einer anderen Tabelle verweist. In diesem Beispiel:

- `rezeptFK` in der Tabelle `Zutat` verweist auf `rezeptPK` in der Tabelle `Rezept`. Dies stellt sicher, dass jede Zutat einem Rezept zugeordnet ist.
- `lagerortFK` in der Tabelle `Zutat` verweist auf `lagerortPK` in der Tabelle `Lagerort`. Dies stellt sicher, dass jede Zutat einem Lagerort zugeordnet ist.

Referenzielle Integrität:

Die referenzielle Integrität ist eine Regel in relationalen Datenbanken, die sicherstellt, dass Fremdschlüsselwerte auf gültige Primärschlüsselwerte verweisen. In diesem Beispiel:

- Der Wert von `rezeptFK` in der Tabelle `Zutat` muss in der Spalte `rezeptPK` der Tabelle `Rezept` vorhanden sein.
- Der Wert von `lagerortFK` in der Tabelle `Zutat` muss in der Spalte `lagerortPK` der Tabelle `Lagerort` vorhanden sein.

Die referenzielle Integrität verhindert Dateninkonsistenzen und stellt sicher, dass die Beziehungen zwischen den Entitäten in der Datenbank korrekt sind.