

Aufgabe 1

Beschreibung des Anwendungsfeldes:

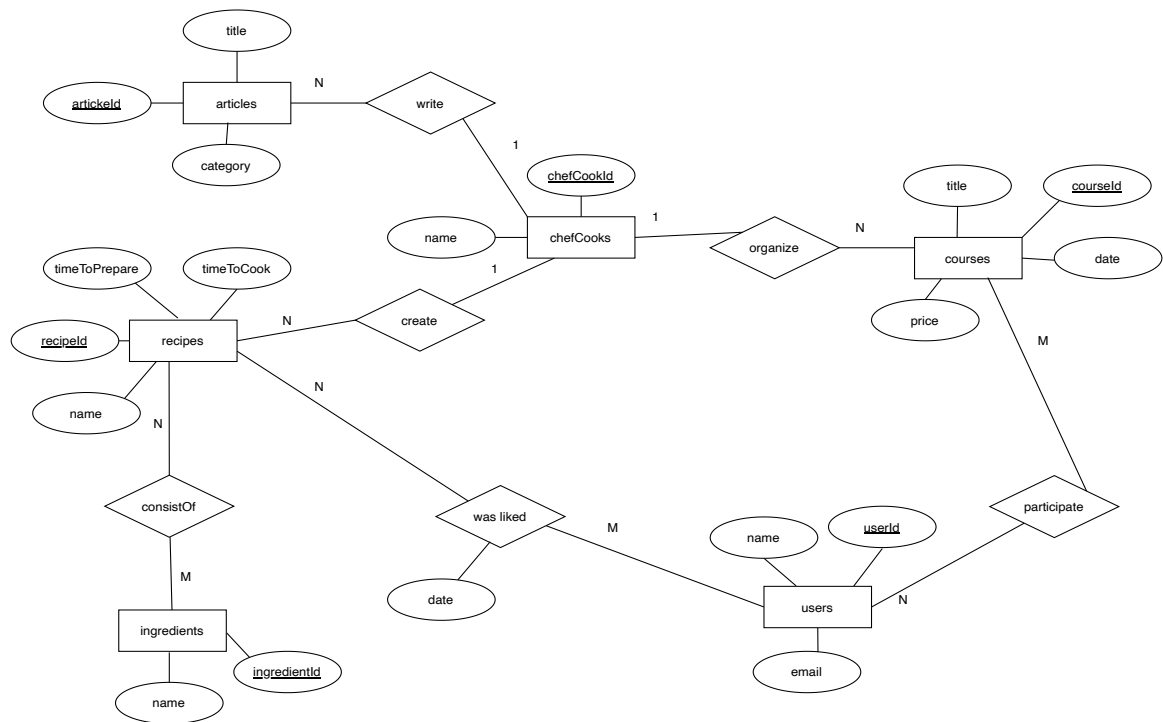
Eine Gruppe von Prominenten Köchen will eine Kochschule öffnen. Alle Informationen sollen digitalisiert und in einer Datenbank gespeichert werden. Jeder Koch soll durch einen Namen und eine einzigartige Nummer beschrieben werden. Die Köche sind sehr kreativ und erschaffen neue Rezepte. Jedes Rezept hat einen Namen, einzigartige Nummer, Vorbereitungszeit, Kochzeit. Natürlich jedes Rezept besteht aus eine Menge der Ingredients. Ingredients haben einen Namen und einzigartige Nummer.

Die Rezepte können von Users gelesen werden. Users sind mit einzigartige Nummer, Name, Email beschrieben werden. Users können die Rezepten, die sie besonders interessant finden mit like markieren, dabei wird Datum gespeichert, wann ein Benutzer ein bestimmtes Rezept geliked hat. Um mehr Users zu gewinnen schreiben die Köche verschiedene interessante und informative Artikels. Artikels werden durch einen Namen, ein Thema, und einzigartige Nummer beschrieben. Die Köche halten regelmäßig Kursen, wo sie zeigen und ausführlich erklären wie ein oder andere Rezept gekocht wird. Users können an den Kursen teilnehmen. Jeder Kurs wird an einem bestimmten Tag durchgeführt also jeder Kurs hat ein Datum. Außerdem wird jeder Kurs durch Preis, Name, und einzigartige Nummer beschreiben.

Aufgabe 2:

Erstellen Sie das konzeptuelle Modell zu Ihrem in Aufgabe 1 beschriebenen Anwendungsfeld:

- a) Erstellen Sie ein ER-Diagramm, wobei Sie die in der Vorlesung vorgestellte Notation benutzen sollen.
- b) Weisen Sie den einzelnen Beziehungen Ihre Funktionalitäten zu.



Aufgabe 3:

Überführen Sie das konzeptuelle Modell in den Implementationsentwurf, wobei Sie das Relationale Datenmodell als Basis verwenden:

- a) Modellieren Sie die für die Umsetzung Ihres konzeptuellen Modells erforderlichen Tabellen. Benutzen Sie dabei die in der Vorlesung vorgestellte Notation und überlegen Sie sich, in welchen Fällen es zu Anomalien kommen könnte und vermeiden Sie diese!
- b) Optimieren Sie Ihr Modell, in dem Sie es vereinfachen und die Anomalien ausschließen.

Relationen aus Entitäten:

recipes: {[recipeId:integer, name: string, timeToPrepare: integer, timeToCook: integer]}

ingredients: {[ingredientId: integer, name: string]}

users: {[userId: integer, name: string, email: string]}

chefCooks: {[chefCookId: integer, name: string]}

courses: {[courseId: integer, title:string, data: date, price: integer]}

articles: {[articleId:integer, title: string, category: string]}

Relationen aus Beziehungen:

organize: {[chefCookId: integer, courseId: integer]}

write: {[chefCookId:integer, articleId:integer]}

create: {[chefCookId:integer, recipeId:integer]}

consistOf: {[recipeId:integer, ingredientId:integer]}

participate: {[userId:integer, courseId:integer]}

wasliked: {[userId:integer, recipeId:integer, date: date]}

Können vereinfacht werden:

~~organize: {[chefCookId: integer, courseId: integer]}~~

~~write: {[chefCookId:integer, articleId:integer]}~~

~~create: {[chefCookId:integer, recipeId:integer]}~~

Alle Relationen Endergebnis:

articles: {[articleId:integer, title:string, category:string, chefCookId:integer]}

courses: {[courseId:integer, title:string, data:date, price:integer,
chefCookId:integer]}

chefCook: {[chefCookId:integer, name:string]}

recipes: {[recipeId:integer, name:string, timeToPrepare:integer,
timeToCook:integer, chefCookId:integer]}

consistOf: {[recipeId:integer, ingredientId:integer]}

ingredients: {[ingredientId:integer, name:string]}

users: {[userId:integer, name:string, email:string]}

participate: {[userId:integer, courseId:integer]}

wasLiked: {[userId:integer, recipeId:integer, date:date]}