CustomCooking

un’applicazione social di ricette che ti aiuta a scegliere cosa cucinare sulla base di quello che hai in casa.

Cognome Nome

Matricola

Email

Cognome Nome

Matricola

Email

Cognome Nome

Matricola

Email

Table of Contents

[**1** Analisi dei requisiti 2](#_Toc408324691)

[*1.1* Requisiti espressi in linguaggio naturale 2](#_Toc408324692)

[*1.2* Glossario dei termini 2](#_Toc408324693)

[*1.3* Eliminazione delle ambiguità presenti 2](#_Toc408324694)

[*1.4* Strutturazione dei requisiti 2](#_Toc408324695)

[*1.5* Specifica operazioni 3](#_Toc408324696)

[**2** Progettazione concettuale 3](#_Toc408324697)

[*2.1* Identificazione delle entità e relazioni 3](#_Toc408324698)

[*2.2* Un primo scheletro dello schema 3](#_Toc408324699)

[*2.3* Sviluppo delle componenti dello scheletro 3](#_Toc408324700)

[*2.4* Unione delle componenti nello schema finale ridotto 3](#_Toc408324701)

[*2.5* Dizionario dei dati 3](#_Toc408324702)

[*2.6* Regole aziendali 4](#_Toc408324703)

[**3** Progettazione logica 5](#_Toc408324704)

[*3.1* Tavole dei volumi e delle operazioni 5](#_Toc408324705)

[*3.2* Ristrutturazione dello schema concettuale 5](#_Toc408324706)

[*3.3* Normalizzazione 5](#_Toc408324707)

[*3.4* Traduzione verso il modello relazionale 5](#_Toc408324708)

[**4** Codifica SQL 5](#_Toc408324709)

[*4.1* Definizione dello schema 5](#_Toc408324710)

[*4.2* Codifica delle operazioni 7](#_Toc408324711)

[**5** Testing 9](#_Toc408324712)

# Analisi dei requisiti

## Requisiti espressi in linguaggio naturale

Si vuole realizzare una base di dati per un’app di ricette.

L’app deve permettere di scegliere una ricetta sulla base delle disponibilità dell’utente, in termini di prodotti alimentari e utensili da cucina.

La base di dati si compone delle entità *Recipe*, *Product*, *Tool*, *Account*, *Step*, e delle relazioni *Ingredient*, *Author*, *Rating*, *ToolAvailability*, *ProductAvailability*, *ToolSet*, *Sequence*.

Per quanto concerne *Recipe*, ci interessa rappresentare un identificatore univoco (idRecipe), nome, tipo, descrizione, cucina a cui appartiene, eventuale origine regionale, tempo stimato di preparazione, difficoltà secondo l’autore, difficoltà secondo le valutazioni degli utenti utilizzatori, eventuale nome e riferimento ad autore esterno della ricetta (ad esempio l’autore del libro di ricette dalla quale la ricetta è stata estrapolata).

Per quanto concerne *Product*, ci interessa rappresentare nome, classe del prodotto, eventuale URL di un’immagine didascalica.

Per quanto concerne *Tool*, ci interessa rappresentare nome ed eventuale URL di un’immagine didascalica.

Per quanto concerne *User*, ci interessa rappresentare username, email e password.

Per quanto concerne *Step*, ci interessa rappresentare un identificatore numerico, descrizione, tempo stimato, eventuale URL di un’immagine didascalica, eventuale URL di un video esplicativo.

## Glossario dei termini

|  |  |  |  |
| --- | --- | --- | --- |
| Termine | Descrizione | Sinonimi | Collegamenti |
| Account | Utente registrato |  | Recipe, ProductAvalaibilty, ToolAvailability, Rating, Author |
| Recipe | Algoritmo grastronomico |  | Account, Step, Tool, Product |
| Product | Prodotto alimentare |  | Ingredients, User,Recipe |
| Tool | Strumento per cucinare |  | Account, Recipe |
| Step | Passo atomico di una ricetta |  | Recipe |
| Ingredient | Prodotto alimentare relativo ad una ricetta |  | Product, Recipe |
| Sequence | Sequenza di step per preparare una ricetta |  | Recipe |
| ProductAvalaibilty | Disponibilità di un prodotto da parte di un utente |  | Account, Product |
| ToolAvailability | Disponibilità di uno strumento da parte di un utente |  | Account, Tool |
| Rating | Valutazione di una ricetta da parte di un utente |  | Account, Recipe |
| Author | Autore di una ricetta |  | Account, Recipe |
| ToolSet | Strumento relativo ad una ricetta |  | Tool, Recipe |

## Eliminazione delle ambiguità presenti

## Strutturazione dei requisiti

* Frasi di carattere generale

Si vuole costruire una base di dati per un’applicazione mobile che consente di scegliere una ricetta sulla base delle proprie disponibilità in termini di prodotti e utensili, filtrando in termini di tempo, difficoltà e valutazione.

* Frasi relative a *Recipe*
* Frasi relative a *Product*
* Frasi relative a *Tool*
* Frasi relative a *Account*
* Frasi relative a *Step*
* Frasi relative a *Recipe*

## Specifica operazioni

* Inserire un nuovo *Account*.
* Inserire una nuova *Recipe*.
* Inserire un nuovo *Step*.
* Inserire un nuovo *Product*.
* Inserire un nuovo *Tool*.
* Aggiungere uno *Step* ad una *Recipe*.
* Aggiungere un *Product* ad una *Recipe*.
* Aggiungere un *Tool* ad una *Recipe*.
* Aggiungere un *Product* ad un *Account*.
* Aggiungere *Tool* ad un *Account*.
* Valutare una *Recipe*.
* Visualizzare le ricette filtrate e ordinate sulla base di parametri specifici.

# Progettazione concettuale

## Identificazione delle entità e relazioni

Applicando la strategia *bottom-*up sono state identificate le seguenti entità e relazioni: Account, Recipe, Product, Ingredient, Tool.

## Un primo scheletro dello schema

A partire da questo elenco è stato modellato il seguento primo scheletro di schema contettuale:

## Sviluppo delle componenti dello scheletro

Ogni ricetta è costituita da uno o più step. Inizialmente si era deciso di dividere la ricetta in due differenti entità:

La nuova relazione (Sequence) è finalizzata a suddividere le ricette in fasi preparative atomiche, affinché queste possano essere condivise, evitando ripetizioni. La nuova entità (Step) rappresenta la singola fase preparativa atomica.

Ogni ricetta è collegata ad un utente che l’ha inserita e creata. Si prevede inoltre la possibilità di inserire ricette create da autori noti.

## Unione delle componenti nello schema finale ridotto

## Dizionario dei dati

Entità:

|  |  |  |  |
| --- | --- | --- | --- |
| Nome entità | Descrizione | Attributi | Identificatore |
| Recipe | Ricetta | idRecipe, nameRecipe, typeRecipe, descriptionRecipe, cuisine, regionalOrigin, preparationTime, difficultyAuthor, difficultyUsers, authorName, authorLink, username, rating | idRecipe |
| Product | Prodotto alimentare | nameProduct, classProduct, unityOfMeasure, subclassProduct, photoProduct | nameProduct |
| Account | Utente registrato | username, pass, email, creationTime | username |
| Tool | Strumento per cucinare | nameTool, photoTool | nameTool |
| Step | Passo esecutivo di una ricetta | idStep, descriptionStep, timeStep, photoStep, videoStep | idStep |

Relazioni:

|  |  |  |  |
| --- | --- | --- | --- |
| Nome relazione | Descrizione | Entità coinvolte | Attributi |
| Rating | Valutazione di una ricetta | Account (0,N) - Recipe(0,N) | username, idRecipe, commentRating, rating, difficulty |
| Ingredient | Ingrediente di una ricetta | Product(1,N) - Recipe(1,N) | idRecipe, nameProduct, quantity, optional |
| Sequence | Passo esecutivo di una ricetta | Recipe(1,N) - Step(1,N) |  |
| ToolSet | Strumento usato in una ricetta | Tool(0,N) - Recipe(0,N) | idRecipe, nameTool, optional |
| ToolAvailability | Strumento nelle disponibilità di un utente | Account (0,N) - Tool(0,N) | username, nameTool |
| ProductAvailability | Prodotto nelle disponibilità di un utente | Account (0,N) - Product(0,N) | username, nameProduct, quantity |
| Author | Autore di una ricetta | User(0,N) - Recipe(0,N) | idRecipe, username |

## 

## Regole aziendali

|  |
| --- |
| Regole di vincolo |
| 1. L’attributo *classProduct* può assumere i seguenti valori: *Meat*, *Vegetable*, *Mushrooms*, *Fruit*, *Fish*, *Spices*, *Cereals*, *Dairy product*, *Alcohol*, *Flour*. 2. L’attributo *typeRecipe* può assumere i seguenti valori: *Starter*, *First course*, *Second course*, *Dessert*, *All in one*. 3. L’attributo *cuisine* può assumere come valore un insieme predefinito ma estendibile di tipi di cucine. 4. L’attributo *regionalOrigin* può assumere come valore un insieme predefinito ma estendibile di regioni geografiche. 5. Il *rating* di una ricetta deve essere compreso tra 1 e 4. 6. La *difficulty* di una ricetta deve essere compresa tra 1 e 4. |
| Regole di derivazione |
| 1. L’attributo *ratingUsers* di una ricetta deve essere calcolato come media dei giudizi espressi dagli utenti per quella ricetta. 2. L’attributo *preparationTime* di una ricetta deve essere calcolato come somma dei tempi stimati per ogni passo della ricetta. |

# Progettazione logica

## Tavole dei volumi e delle operazioni

## Ristrutturazione dello schema concettuale

## Normalizzazione

## Traduzione verso il modello relazionale

# Codifica SQL

## Definizione dello schema

1. **create** **database** if not exists CustomCooking;
2. use CustomCooking;
4. **drop** **table** if exists Rating;
5. **drop** **table** if exists ToolAvailability;
6. **drop** **table** if exists ProductAvailability;
7. **drop** **table** if exists Ingredient;
8. **drop** **table** if exists ToolSet;
9. **drop** **table** if exists Author;
10. **drop** **table** if exists Sequence;
11. **drop** **table** if exists Step;
12. **drop** **table** if exists Recipe;
13. **drop** **table** if exists Product;
14. **drop** **table** if exists Tool;
15. **drop** **table** if exists Account;
17. **create** **table** Account (
18. username **varchar**(30) not null,
19. pass **varchar**(32) not null,
20. email **varchar**(100),
21. creationTime **datetime**,
23. **primary** **key**(username)
24. );
26. **create** **table** Recipe (
27. idRecipe **int** not null **auto\_increment**,
28. nameRecipe **varchar**(30) not null,
29. typeRecipe **varchar**(30) not null,
30. descriptionRecipe text,
31. cuisine **varchar**(30),
32. regionalOrigin **varchar**(30),
33. preparationTime **int** not null,
34. difficultyAuthor **int** not null,
35. difficultyUsers **double**,
36. authorName **varchar**(30),
37. authorLink **varchar**(300),
38. username **varchar**(30) not null,
39. rating **double**,
41. **primary** **key**(idRecipe),
42. **foreign** **key**(username) **references** Account(username)
43. );
45. **create** **table** Step (
46. idStep **int** not null **auto\_increment**,
47. descriptionStep **varchar**(300) not null,
48. timeStep **int** not null,
49. photoStep **varchar**(200),
50. videoStep **varchar**(200),
52. **primary** **key**(idStep)
53. );
55. **create** **table** Product (
56. nameProduct **varchar**(30) not null,
57. classProduct **varchar**(30) not null,
58. unityOfMeasure **varchar**(10) not null,
59. subclassProduct **varchar**(30),
60. photoProduct **varchar**(200),
62. **primary** **key**(nameProduct)
63. );
65. **create** **table** Tool (
66. nameTool **varchar**(30) not null,
67. photoTool **varchar**(200),
69. **primary** **key**(nameTool)
70. );
72. **create** **table** Ingredient (
73. idRecipe **int** not null,
74. nameProduct **varchar**(30) not null,
75. quantity **double** not null,
76. optional boolean not null **default** **false**,
78. **primary** **key**(idRecipe, nameProduct),
79. **foreign** **key**(idRecipe) **references** Recipe(idRecipe),
80. **foreign** **key**(nameProduct) **references** Product(nameProduct)
81. );
83. **create** **table** ProductAvailability (
84. username **varchar**(30) not null,
85. nameProduct **varchar**(30) not null,
86. quantity **double** not null,
88. **primary** **key**(username, nameProduct),
89. **foreign** **key**(username) **references** Account(username),
90. **foreign** **key**(nameProduct) **references** Product(nameProduct)
91. );
93. **create** **table** ToolAvailability (
94. username **varchar**(30) not null,
95. nameTool **varchar**(30) not null,
97. **primary** **key**(username, nameTool),
98. **foreign** **key**(username) **references** Account(username),
99. **foreign** **key**(nameTool) **references** Tool(nameTool)
100. );
102. **create** **table** Rating (
103. username **varchar**(30) not null,
104. idRecipe **int** not null,
105. commentRating text,
106. rating **int** not null,
107. difficulty **int**,
109. **primary** **key**(username, idRecipe),
110. **foreign** **key**(username) **references** Account(username),
111. **foreign** **key**(idRecipe) **references** Recipe(idRecipe)
112. );
114. **create** **table** Sequence (
115. idRecipe **int** not null,
116. idStep **int** not null,
117. position **int** not null,
119. **primary** **key**(idRecipe, idStep),
120. **foreign** **key**(idRecipe) **references** Recipe(idRecipe),
121. **foreign** **key**(idStep) **references** Step(idStep)
122. );
124. **create** **table** ToolSet (
125. idRecipe **int** not null,
126. nameTool **varchar**(30) not null,
127. optional boolean not null,
129. **primary** **key**(idRecipe, nameTool),
130. **foreign** **key**(idRecipe) **references** Recipe(idRecipe),
131. **foreign** **key**(nameTool) **references** Tool(nameTool)
132. );
134. **create** **table** Author (
135. idRecipe **int** not null,
136. username **varchar**(30) not null,
138. **primary** **key**(idRecipe, username),
139. **foreign** **key**(idRecipe) **references** Recipe(idRecipe),
140. **foreign** **key**(username) **references** Account(username)

## 

## Codifica delle operazioni

1. /\* Stored Procedures \*/
2. DELIMITER $
4. **drop** **procedure** if exists addAuthor $
5. **create** **procedure** addAuthor(in username **varchar**(30), pass **varchar**(30), email **varchar**(30))
6. comment 'Add an author'
7. **begin**
8. **insert** **into** Account **values** (username, md5(pass), email, now());
9. **end** $
11. **drop** **procedure** if exists addProduct $
12. **create** **procedure** addProduct(in nameProduct **varchar**(30), classProduct **varchar**(30), unityOfMeasure **varchar**(10), subclassProduct **varchar**(30), photoProduct **varchar**(200))
13. comment 'Add a product'
14. **begin**
15. **insert** **into** Product **values** (nameProduct, classProduct, unityOfMeasure, subclassProduct, photoProduct);
16. **end** $
18. **drop** **procedure** if exists addTool $
19. **create** **procedure** addTool(in nameTool **varchar**(30), photoTool **varchar**(200))
20. comment 'Add a tool'
21. **begin**
22. **insert** **into** Tool **values** (nameTool, photoTool);
23. **end** $
25. **drop** **procedure** if exists addIngredient $
26. **create** **procedure** addIngredient(in idRecipe **integer**, nameProduct **varchar**(30), quantity **double**, optional boolean)
27. comment 'Add an ingredient'
28. **begin**
29. **insert** **into** Ingredient **values** (idRecipe, nameProduct, quantity, optional);
30. **end** $
32. **drop** **procedure** if exists addAvailability $
33. **create** **procedure** addAvailability(in username **varchar**(30), nameProduct **varchar**(30), quantity **double**)
34. comment 'Add an availability'
35. **begin**
36. **insert** **into** ProductAvailability **values** (username, nameProduct, quantity);
37. **end** $
39. **drop** **procedure** if exists insertStep $
40. **create** **procedure** insertStep(in idRecipe **integer**, idStep **integer**, position **integer**)
41. comment 'Insert a step in a sequence'
42. **begin**
43. **insert** **into** **Sequence** **values** (idRecipe, idStep, position);
44. **end** $
46. **drop** **procedure** if exists createRecipe $
47. **create** **procedure** createRecipe(in nameRecipe **varchar**(30), typeRecipe **varchar**(30), descriptionRecipe text,
48. cuisine **varchar**(30), regionalOrigin **varchar**(30), difficulty **integer**, username **varchar**(30))
49. comment 'Create a recipe'
50. **begin**
51. **insert** **into** Recipe (nameRecipe, typeRecipe, descriptionRecipe, cuisine, regionalOrigin, preparationTime, difficultyAuthor, username)
52. **values** (nameRecipe, typeRecipe, descriptionRecipe, cuisine, regionalOrigin, 0, difficultyAuthor, username);
53. **end** $
55. **drop** **procedure** if exists rateRecipe $
56. **create** **procedure** rateRecipe(in username **varchar**(30), idRecipe **int**, commentRating text, rating **int**, difficulty **int**)
57. comment 'Rate a recipe'
58. **begin**
59. **insert** **into** Rating **values** (username, idRecipe, commentRating, rating, difficulty);
60. **end** $
62. **drop** **procedure** if exists createStep $
63. **create** **procedure** createStep(in descriptionStep text, timeStep **int**, photoStep **varchar**(200), videoStep **varchar**(200))
64. comment 'Create a step'
65. **begin**
66. **insert** **into** Step (descriptionStep, timeStep, photoStep, videoStep) **values** (descriptionStep, timeStep, photoStep, videoStep);
67. **end** $
69. **drop** **procedure** if exists getRecipes $
70. **create** **procedure** getRecipes(in timeMin **int**, timeMax **int**, diffMin **int**, diffMax **int**, username **varchar**(30))
71. comment 'Get the possible recipes for a given user accordingly by time, difficulty and availability, ordered by rating'
72. **begin**
73. **select** Recipe.\*
74. **from** Recipe
75. **where**
76. preparationTime between timeMin and timeMax and
77. (
78. (difficultyAuthor between diffMin and diffMax) or
79. (difficultyUsers between diffMin and diffMax)
80. ) and
81. idRecipe not in (
82. **select** idRecipe
83. **from** Ingredient i
84. **where**
85. i.optional = **false** and
86. not exists (
87. **select** \*
88. **from** ProductAvailability a
89. **where**
90. a.username = username and
91. i.nameProduct = a.nameProduct and
92. i.quantity <= a.quantity))
93. **order** **by** rating **desc**;
94. **end** $
96. /\* Automatically update the overall rating of a recipe. \*/
97. **drop** **trigger** if exists updateRating $
98. **create** **trigger** updateRating
99. **after** **insert** **on** Rating
100. **for** each row
101. **begin**
102. **declare** overallRating **double**;
103. **declare** overallDifficulty **double**;
105. **select** avg(rating) **into** overallRating
106. **from** Rating
107. **where** idRecipe = new.idRecipe;
109. **update** Recipe
110. **set** rating = overallRating
111. **where** idRecipe = new.idRecipe;
113. if (new.difficulty != null) **then**
114. **select** avg(difficulty) **into** overallDifficulty
115. **from** Rating
116. **where** idRecipe = new.idRecipe;
118. **update** Recipe
119. **set** difficultyUsers = overallDifficulty
120. **where** idRecipe = new.idRecipe;
121. **end** if;
122. **end** $
124. /\* Automatically update the overall time needed for a recipe. \*/
125. **drop** **trigger** if exists updateTime $
126. **create** **trigger** updateTime
127. **after** **insert** **on** **Sequence**
128. **for** each row
129. **begin**
130. **declare** overallTime **int**;
132. **select** sum(timeStep) **into** overallTime
133. **from** **Sequence** natural join Step
134. **where** idRecipe = new.idRecipe;
136. **update** Recipe
137. **set** preparationTime = overallTime
138. **where** idRecipe = new.idRecipe;
139. **end** $
141. DELIMITER ;

# Testing

1. /\* Testing \*/
2. call createAuthor("tommy", "password1", "tommy@studio.unibo.it");
3. call createAuthor("alex", " password2", "alex@studio.unibo.it");
5. **select** \* **from** Account;
7. call createRecipe('Sushi', 'I course', 'Rice and fish', 'Japonese', null, 2, 'tommy');
8. call createRecipe('Ceasar Salad', 'III course', null, 'American', null, 1, 'alex');
9. call createRecipe('Seitan', 'II course', 'A vegan "meat" made with soy', null, null, 3,'tommy');
11. call rateRecipe("alex", 1, "A wonderful dish!", 4, 1);
13. **select** \* **from** Recipe **where** idRecipe = 1;
15. call rateRecipe("tommy", 1, "A dreadful dish!", 1, 3);
17. **select** \* **from** Recipe **where** idRecipe = 1;
19. call getRecipes(0, 40, 0, 4, "alex");
21. call createStep('Wash the rice many times', 10, null, null);
22. call createStep('Boil the water', 12, null, null);
23. call createStep('Cut the vegetables at Julienne', 5, null, null);
25. **select** preparationTime **from** Recipe **where** idRecipe = 1;
27. call insertStep(1, 1, 1);
28. call insertStep(1, 2, 2);
30. **select** preparationTime **from** Recipe **where** idRecipe = 1;
32. call addProduct('Rice', 'Cereal', 'gr', null, null);
33. call addProduct('Cocumber', 'Vegetable', 'gr', null, null);
34. call addProduct('Salmon roe', 'Fish', 'gr', null, null);
36. call addTool('hagiri',null);
38. call addIngredient(1, 'Rice', 100, **false**);
39. call addIngredient(1, 'Cocumber', 3, **false**);
40. call addIngredient(1, 'Salmon roe', 25, **true**);
42. call addAvailability('alex', 'Rice', 100);
43. call addAvailability('alex', 'Cocumber', 4);
44. call addAvailability('tommy', 'Rice', 500);