Food Network Recipes Database
Database Design Specifications

By: Jessica Rieger
# Table of Contents

**Executive Summary** ................................................................. 4  
Overview .................................................................................. 4  
Objectives ............................................................................... 4  

**Entity-Relationship Diagram** .................................................. 6  

**Tables** .................................................................................. 7  
People ....................................................................................... 7  
Chefs ......................................................................................... 8  
Authors ...................................................................................... 9  
CookbookAuthors ...................................................................... 10  
BlogAuthors ............................................................................. 11  
PublishMethods ........................................................................ 12  
Published .................................................................................. 13  
Blogs ......................................................................................... 14  
Cookbooks ............................................................................... 15  
Recipes ..................................................................................... 16  
RecipeGroups ........................................................................... 17  
Utensils ..................................................................................... 18  
UtensilsList ............................................................................... 19  
Photos ....................................................................................... 20  
PhotosList ................................................................................ 21  
Steps .......................................................................................... 22  
Ingredients ............................................................................... 23  
IngredientsList .......................................................................... 24  
Parts ......................................................................................... 25  
PartsList .................................................................................... 26  
Substitutions ............................................................................. 27  
IngredientsofSubstitution ......................................................... 28  
FlavorAffinities ........................................................................ 29  
AffinitiesList ............................................................................ 30  
Themes ....................................................................................... 31  
RecipeThemesList ...................................................................... 32  
PublishMethodsThemesList ....................................................... 33  

**Views** .................................................................................... 34  
EntireRecipe .............................................................................. 34  
IngredientAffinities ................................................................. 35  
PublicationInformation .............................................................. 36  

**Reports** ................................................................................. 37  
Number of Recipes Published by an Author in a Particular Year ......................................................... 37  
Average Complexity of Recipes in the Database .................................................................................. 38  
Theme Usage ........................................................................... 39  
Potential Tweaks to a Recipe ...................................................... 40  

**Stored Procedures** ................................................................. 41  
Get_Recipe_Ingredientslist_byName ......................................................................................... 41
Executive Summary

Overview

The world of food is expanding in every direction. Each day thousands of new ingredients, utensils, flavor combos, substitutions, chefs, and recipes are emerging. Recording and managing this huge repository of information is a challenge that many companies in the food industry are facing.

In order to address this problem, companies must find a way to organize the information so that it can be accessed quickly to suit their needs. However, the complexities of the relationships between recipes, their source of publications, and all the different flavor affinities can be daunting. To stay on top of all the new information, a company must have a solid foundation for storage that works no matter what new flavor combo or publication method the chefs and authors of the world decide to invent.

Objectives

The purpose of this document is to outline a database system to manage Food Network’s records of recipes and their related information. This includes their authors and chefs, their source of publication, themes associated with the recipe and publication source, substitutions for different ingredients in the recipe, possible ways to tweak the recipes based on their ingredients’ membership in different flavor affinities, and information about the recipes’ ingredients, utensils required, and photos. This database is intended to provide a way for the company to manage the huge amounts of data that are associated with recipe storing.
This document will provide an overview of the database and its technical and implementation details. It will outline the tables and their functional dependencies, views, reports, stored procedures, triggers, and security features. It will also explore the database's potential for expansion and enhancements.
Entity-Relationship Diagram
Tables

People

Purpose

This table is used to store the ID, first name, last name and date of birth of the people associated with the database including chefs, cookbook authors, and blog authors.

Create Statement

CREATE TABLE People (  
  PerID text not null,  
  Fname text not null,  
  Lname text not null,  
  Birthday date not null,  
  PRIMARY KEY (PerID)  
);

Functional Dependencies

PerID -> Fname, Lname, Birthday

Sample Data

<table>
<thead>
<tr>
<th>PerID</th>
<th>Fname</th>
<th>Lname</th>
<th>Birthday</th>
</tr>
</thead>
<tbody>
<tr>
<td>PER09</td>
<td>Jessica</td>
<td>Rieger</td>
<td>1997-03-23</td>
</tr>
<tr>
<td>PER01</td>
<td>Kathryn</td>
<td>Smith</td>
<td>1990-01-15</td>
</tr>
<tr>
<td>PER02</td>
<td>Ina</td>
<td>Garten</td>
<td>1948-02-02</td>
</tr>
<tr>
<td>PER03</td>
<td>Andie</td>
<td>Mitchell</td>
<td>1985-03-20</td>
</tr>
<tr>
<td>PER04</td>
<td>Tessa</td>
<td>Bramley</td>
<td>1985-01-01</td>
</tr>
<tr>
<td>PER05</td>
<td>Rick</td>
<td>Bayless</td>
<td>1953-11-23</td>
</tr>
<tr>
<td>PER06</td>
<td>Deann</td>
<td>Groen Bayless</td>
<td>1948-10-30</td>
</tr>
<tr>
<td>PER07</td>
<td>Natalie</td>
<td>Smith</td>
<td>1953-11-23</td>
</tr>
<tr>
<td>PER08</td>
<td>Jane</td>
<td>Moorhead</td>
<td>1895-11-23</td>
</tr>
<tr>
<td>PER10</td>
<td>Peggy</td>
<td>Wilson</td>
<td>1896-11-23</td>
</tr>
<tr>
<td>PER11</td>
<td>Lidia</td>
<td>Bastianich</td>
<td>1947-02-21</td>
</tr>
<tr>
<td>PER12</td>
<td>Cassidy</td>
<td>Mazelin</td>
<td>1997-05-21</td>
</tr>
</tbody>
</table>
Chefs

Purpose

This table is used to store the information about a person in regards to their status as a chef, for example their title and current job.

Create Statement

CREATE TABLE Chefs (  
    PerID text not null references People(PerID),  
    Title text,  
    CurrentJob text,  
    PRIMARY KEY (PerID)  
);

Functional Dependencies

PerID -> Title, CurrentJob

Sample Data

<table>
<thead>
<tr>
<th>perid</th>
<th>title text</th>
<th>currentjob text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PER09 Chef (The Gourmet College Chef)</td>
<td>Home Chef</td>
</tr>
<tr>
<td>2</td>
<td>PER12 Sous Chef</td>
<td>The Gourmet College Chefs Sous Chef</td>
</tr>
<tr>
<td>3</td>
<td>PER02 Chef</td>
<td>TV Chef</td>
</tr>
<tr>
<td>4</td>
<td>PER03 Chef in training</td>
<td>Home Chef</td>
</tr>
<tr>
<td>5</td>
<td>PER04 Sous Chef</td>
<td>Pastabilities</td>
</tr>
<tr>
<td>6</td>
<td>PER11 Chef</td>
<td>TV Chef</td>
</tr>
</tbody>
</table>
**Authors**

**Purpose**

This table is used to store the information about a person in regards to their status as an author, for example their source of inspiration.

**Create Statement**

```sql
CREATE TABLE Authors (
    PerID text not null references People(PerID),
    Inspiration text,
    PRIMARY KEY (PerID)
);
```

**Functional Dependencies**

PerID -> Inspiration

**Sample Data**

<table>
<thead>
<tr>
<th>perid</th>
<th>inspiration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PER09 All different types of cuisine!</td>
</tr>
<tr>
<td>2</td>
<td>PER01 Healthy</td>
</tr>
<tr>
<td>3</td>
<td>PER02 Home Cooking</td>
</tr>
<tr>
<td>4</td>
<td>PER03 Healthy and Quick</td>
</tr>
<tr>
<td>5</td>
<td>PER04 Vegetarian</td>
</tr>
<tr>
<td>6</td>
<td>PER05 Mexican Cuisine</td>
</tr>
<tr>
<td>7</td>
<td>PER06 Mexican Cuisine</td>
</tr>
<tr>
<td>8</td>
<td>PER07 Favorite Foods</td>
</tr>
<tr>
<td>9</td>
<td>PER08 Her Grandmas Recipes</td>
</tr>
<tr>
<td>10</td>
<td>PER10 Her Grandmas Recipes</td>
</tr>
<tr>
<td>11</td>
<td>PER11 Italian Cuisine</td>
</tr>
</tbody>
</table>
**CookbookAuthors**

**Purpose**

This table is used to store the information about a person in regards to their status as cookbook author, for example the number of cookbooks they have authored.

**Create Statement**

CREATE TABLE CookbookAuthors (  
  PerID text not null references Authors(PerID),  
  NumCookbooks int,  
  PRIMARY KEY (PerID)  
);

**Functional Dependencies**

PerID -> NumCookbooks

**Sample Data**

<table>
<thead>
<tr>
<th>perid text</th>
<th>numcookbooks integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PER02 4</td>
</tr>
<tr>
<td>2</td>
<td>PER04 2</td>
</tr>
<tr>
<td>3</td>
<td>PER05 1</td>
</tr>
<tr>
<td>4</td>
<td>PER06 7</td>
</tr>
<tr>
<td>5</td>
<td>PER08 1</td>
</tr>
<tr>
<td>6</td>
<td>PER10 1</td>
</tr>
<tr>
<td>7</td>
<td>PER11 30</td>
</tr>
</tbody>
</table>
BlogAuthors

Purpose

This table is used to store the information about a person in regards to their status as a blog author, for example how frequently they post.

Create Statement

CREATE TABLE BlogAuthors (  
   PerID text not null references Authors(PerID),  
   FreqOfPost text,  
   PRIMARY KEY (PerID)  
);

Functional Dependencies

PerID -> FreqOfPost

Sample Data

<table>
<thead>
<tr>
<th>perid</th>
<th>freqofpost</th>
</tr>
</thead>
<tbody>
<tr>
<td>PER09</td>
<td>Once a Week</td>
</tr>
<tr>
<td>PER01</td>
<td>Everyday</td>
</tr>
<tr>
<td>PER03</td>
<td>Once a Week</td>
</tr>
<tr>
<td>PER07</td>
<td>Once a week</td>
</tr>
</tbody>
</table>
PublishMethods

Purpose

This table is used to store the information about publications, in particular cookbooks or blogs. It stores information such as the year it was published/started, the title, and a short description of the content.

Create Statement

CREATE TABLE PublishMethods (
  PID text not null,
  YearPublished text not null,
  Title text not null,
  ContentDescr text,
  PRIMARY KEY (PID)
);

Functional Dependencies

PID -> YearPublished, Title, ContentDescr

Sample Data

<table>
<thead>
<tr>
<th>pid</th>
<th>yearpublished</th>
<th>title</th>
<th>contentdescr</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01</td>
<td>2010</td>
<td>COOKIE &amp; kate</td>
<td>Healthy, simple cooking that has recipes for any time of day.</td>
</tr>
<tr>
<td>P08</td>
<td>2016</td>
<td>The Gourmet College Chef</td>
<td>Whatever I feel like cooking.</td>
</tr>
<tr>
<td>P02</td>
<td>2003</td>
<td>Barefoot Contessa at Home</td>
<td>Easy, simple, home cooking.</td>
</tr>
<tr>
<td>P03</td>
<td>2010</td>
<td>Andie Mitchell</td>
<td>Recipes * Inspiration * Life</td>
</tr>
<tr>
<td>P04</td>
<td>2007</td>
<td>Easy Vegetarian</td>
<td>Simple recipes for lunch, brunch, and dinner.</td>
</tr>
<tr>
<td>P05</td>
<td>1987</td>
<td>Authentic Mexican: Regional Cooking from the Heart of Mexico</td>
<td>Classic Mexican Cuisine.</td>
</tr>
<tr>
<td>P06</td>
<td>2014</td>
<td>Ce que Jaime</td>
<td>Indulgence Foods</td>
</tr>
<tr>
<td>P07</td>
<td>1915</td>
<td>Peoria Womens Cook Book</td>
<td>Your Grandmas Recipes</td>
</tr>
<tr>
<td>P09</td>
<td>2013-10-15</td>
<td>Lidias Commonsense Italian Cooking</td>
<td>Easy, simple, Italian cooking.</td>
</tr>
<tr>
<td>P10</td>
<td>2013</td>
<td>Lidia Recipe Archives</td>
<td>Italian cooking.</td>
</tr>
</tbody>
</table>
Published

Purpose

This table is used to store information about who the authors are of the cookbooks and blogs. In order to have comprehensive information about a publication, it is necessary to know who authored it, thus there is a clear need for this table.

Create Statement

CREATE TABLE Published (  
    PID text not null references PublishMethods(PID),  
    PerID text not null references Authors(PerID),  
    PRIMARY KEY (PerID, PID)  
);

Functional Dependencies

PID, PerID ->

Sample Data

<table>
<thead>
<tr>
<th>pid</th>
<th>perid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P01 PER01</td>
</tr>
<tr>
<td>2</td>
<td>P08 PER09</td>
</tr>
<tr>
<td>3</td>
<td>P02 PER02</td>
</tr>
<tr>
<td>4</td>
<td>P03 PER03</td>
</tr>
<tr>
<td>5</td>
<td>P04 PER04</td>
</tr>
<tr>
<td>6</td>
<td>P05 PER05</td>
</tr>
<tr>
<td>7</td>
<td>P05 PER06</td>
</tr>
<tr>
<td>8</td>
<td>P06 PER07</td>
</tr>
<tr>
<td>9</td>
<td>P07 PER06</td>
</tr>
<tr>
<td>10</td>
<td>P07 PER10</td>
</tr>
</tbody>
</table>
Blogs

Purpose

This table is used to store information about a publication in regards to its status as a blog, for example its URL so that it can be visited if necessary.

Create Statement

CREATE TABLE Blogs (  
    PID text not null references PublishMethods(PID),  
    URL text not null,  
    PRIMARY KEY (PID)  
);  

Functional Dependencies

PID -> URL

Sample Data

<table>
<thead>
<tr>
<th>pid</th>
<th>url</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://cookieandkate.com">http://cookieandkate.com</a></td>
</tr>
<tr>
<td>2</td>
<td><a href="http://thegourmetcollegechef.weebly.com/">http://thegourmetcollegechef.weebly.com/</a></td>
</tr>
<tr>
<td>3</td>
<td><a href="http://andlemitchell.com">http://andlemitchell.com</a></td>
</tr>
</tbody>
</table>
Cookbooks

Purpose

This table is used to store more detailed information about a publication in regards to its status as a cookbook, for example the number of recipes it contains, its price in US dollars, and its publishing company.

Create Statement

CREATE TABLE Cookbooks (  
  PID text not null references PublishMethods(PID),  
  NumRecipes integer,  
  PriceUSD decimal,  
  Publisher text,  
  PRIMARY KEY (PID)  
);

Functional Dependencies

PID -> NumRecipes, PriceUSD, Publisher

Sample Data

<table>
<thead>
<tr>
<th>pid</th>
<th>numrecipes</th>
<th>priceusd</th>
<th>publisher</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>integer</td>
<td>numeric</td>
<td>text</td>
</tr>
<tr>
<td>1</td>
<td>P02</td>
<td>150</td>
<td>18.99 Clarkson Potter/Publishers</td>
</tr>
<tr>
<td>2</td>
<td>P04</td>
<td>75</td>
<td>13.19 Ryland Peters &amp; Small, Inc.</td>
</tr>
<tr>
<td>3</td>
<td>P05</td>
<td>200</td>
<td>26.99 Clarkson Potter/Publishers</td>
</tr>
<tr>
<td>4</td>
<td>P07</td>
<td>170</td>
<td>10.00 J.W. Franks and Sons Printers</td>
</tr>
</tbody>
</table>
Recipes

Purpose

This table is used to store the general information about a recipe, for example its name, creation date, the date it was updated, its prep time, cook time, total time, the number of people it serves, the category that it belongs to, and its source. It also contains the ID of the recipe that can be used to retrieve the ingredients list, steps, utensils list, and photographs associated with the recipe. Its source indicates whether it is original, or it is inspired by another recipe.

Create Statement

CREATE TABLE Recipes (
  RID text not null,
  Name text not null,
  DateCreated date not null,
  DateUpdated date not null,
  PrepTimeMin integer not null,
  CookTimeMin integer not null,
  TotalTimeMin integer,
  Serves integer,
  Category text,
  SRC text not null,
  PRIMARY KEY (RID)
);

Functional Dependencies

RID -> Name, DateCreated, DateUpdated, PrepTimeMin, CookTimeMin,
TotalTimeMin, Serves, Category, SRC

Sample Data

<table>
<thead>
<tr>
<th>rID</th>
<th>name text</th>
<th>datecreated</th>
<th>dateupdated</th>
<th>preptimeMin</th>
<th>cooktimeMin</th>
<th>totaltimeMin</th>
<th>serves</th>
<th>category</th>
<th>SRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Butternut Squash Soup</td>
<td>2015-11-11</td>
<td>2015-11-11</td>
<td>10</td>
<td>55</td>
<td>65</td>
<td>4</td>
<td>Soup</td>
<td>Original</td>
</tr>
<tr>
<td>2</td>
<td>Lemon Cake</td>
<td>2003-01-25</td>
<td>2016-11-11</td>
<td>30</td>
<td>60</td>
<td>90</td>
<td>12</td>
<td>Cake</td>
<td>Original</td>
</tr>
<tr>
<td>3</td>
<td>Slow Cooker Mexican Pulled Pork Tacos</td>
<td>2011-03-08</td>
<td>2016-11-24</td>
<td>15</td>
<td>480</td>
<td>495</td>
<td>4</td>
<td>Tacos</td>
<td>Original</td>
</tr>
<tr>
<td>4</td>
<td>Fiorentina</td>
<td>2007-01-01</td>
<td>2011-11-20</td>
<td>30</td>
<td>20</td>
<td>50</td>
<td>4</td>
<td>Pizza</td>
<td>Original</td>
</tr>
<tr>
<td>5</td>
<td>Cold Chicken and Avocado with Chipotle</td>
<td>1987-01-01</td>
<td>2016-11-11</td>
<td>30</td>
<td>85</td>
<td>105</td>
<td>4</td>
<td>Chicken</td>
<td>Original</td>
</tr>
<tr>
<td>6</td>
<td>Bacon, Brie, and Avocado Sandwich</td>
<td>2014-09-15</td>
<td>2016-11-11</td>
<td>15</td>
<td>10</td>
<td>25</td>
<td>2</td>
<td>Sandwich</td>
<td>Original</td>
</tr>
<tr>
<td>7</td>
<td>Steak A La Creole</td>
<td>1915-09-15</td>
<td>2016-11-11</td>
<td>25</td>
<td>120</td>
<td>145</td>
<td>4</td>
<td>Steak</td>
<td>Original</td>
</tr>
<tr>
<td>8</td>
<td>Mexican Chili</td>
<td>1915-09-15</td>
<td>2016-11-12</td>
<td>20</td>
<td>30</td>
<td>50</td>
<td>4</td>
<td>Chili</td>
<td>Original</td>
</tr>
<tr>
<td>9</td>
<td>Chocolate Chip Cookies</td>
<td>1915-09-15</td>
<td>2016-11-12</td>
<td>140</td>
<td>12</td>
<td>152</td>
<td>20</td>
<td>Cookies</td>
<td>My Mother</td>
</tr>
<tr>
<td>10</td>
<td>Almond and Coffee Cream Mini-Tarts</td>
<td>2013-10-15</td>
<td>2016-11-12</td>
<td>75</td>
<td>25</td>
<td>100</td>
<td>8</td>
<td>Tarts</td>
<td>Original</td>
</tr>
</tbody>
</table>
**RecipeGroups**

**Purpose**

This table allows us to know which recipes are in each publication method.

**Create Statement**

```sql
CREATE TABLE RecipeGroups (  
    PID text not null references PublishMethods(PID),  
    RID text not null references Recipes(RID),  
 PRIMARY KEY (PID, RID)  
);
```

**Functional Dependencies**

PID, RID ->

**Sample Data**

<table>
<thead>
<tr>
<th>pid</th>
<th>rid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P01 R01</td>
</tr>
<tr>
<td>2</td>
<td>P08 R01</td>
</tr>
<tr>
<td>3</td>
<td>P02 R02</td>
</tr>
<tr>
<td>4</td>
<td>P08 R02</td>
</tr>
<tr>
<td>5</td>
<td>P03 R03</td>
</tr>
<tr>
<td>6</td>
<td>P08 R03</td>
</tr>
<tr>
<td>7</td>
<td>P04 R04</td>
</tr>
<tr>
<td>8</td>
<td>P08 R04</td>
</tr>
<tr>
<td>9</td>
<td>P05 R05</td>
</tr>
<tr>
<td>10</td>
<td>P06 R06</td>
</tr>
<tr>
<td>11</td>
<td>P08 R06</td>
</tr>
<tr>
<td>12</td>
<td>P07 R07</td>
</tr>
<tr>
<td>13</td>
<td>P07 R08</td>
</tr>
<tr>
<td>14</td>
<td>P08 R09</td>
</tr>
<tr>
<td>15</td>
<td>P09 R10</td>
</tr>
<tr>
<td>16</td>
<td>P10 R10</td>
</tr>
</tbody>
</table>
Utensils

Purpose

This table is used to store information about utensils, such as their name, the recommended brand, and the material that they are made of. It also stores an alternate utensil that can be used if the suggested utensil is unavailable. This is useful as the perfect utensil is not always readily available in the chef's kitchen.

Create Statement

CREATE TABLE Utensils (  
    UID text not null,  
    Name text not null,  
    Brand text,  
    AlternateUtensil text,  
    Material text,  
    PRIMARY KEY (UID)  
);

Functional Dependencies

UID -> Name, Brand, AlternateUtensil, Material

Sample Data

<table>
<thead>
<tr>
<th>uid</th>
<th>name</th>
<th>brand</th>
<th>alternateutensil</th>
<th>material</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U01 High-Performance Blender</td>
<td>Vitamix</td>
<td>Immersion Blender</td>
<td>n/a</td>
</tr>
<tr>
<td>2</td>
<td>U02 8 1/2 by 4 1/4 by 2 1/2 inch loaf pan</td>
<td>William Sonoma</td>
<td>alternate size loaf pans</td>
<td>aluminum</td>
</tr>
<tr>
<td>3</td>
<td>U03 Crockpot</td>
<td>William Sonoma</td>
<td>None</td>
<td>Clay</td>
</tr>
<tr>
<td>4</td>
<td>U04 Large Saucepan</td>
<td>All-Clad</td>
<td>Pasta Pot</td>
<td>Calphalon</td>
</tr>
<tr>
<td>5</td>
<td>U05 Strainer</td>
<td>None</td>
<td>None</td>
<td>Any</td>
</tr>
<tr>
<td>6</td>
<td>U06 Pizza Stone</td>
<td>Old Stone</td>
<td>Baking Sheet</td>
<td>Stone</td>
</tr>
<tr>
<td>7</td>
<td>U07 Medium Saucepan</td>
<td>William Sonoma</td>
<td>Pasta Pot</td>
<td>Calphalon</td>
</tr>
<tr>
<td>8</td>
<td>U08 Baking Sheet</td>
<td>Nordic Ware</td>
<td>aluminum foil</td>
<td>aluminum</td>
</tr>
<tr>
<td>9</td>
<td>U09 Large Knife</td>
<td>William Sonoma</td>
<td>Medium Knife</td>
<td>Stainless Steel</td>
</tr>
<tr>
<td>10</td>
<td>U10 Medium skillet</td>
<td>Cuisinart</td>
<td>Large skillet</td>
<td>Stainless Steel</td>
</tr>
</tbody>
</table>
**UtensilsList**

**Purpose**

This table is used to store information about which utensils are needed for a particular recipe as well as the number of each utensil required.

**Create Statement**

CREATE TABLE UtensilsList (  
    RID text not null references Recipes(RID),  
    UID text not null references Utensils(UID),  
    Quantity integer not null,  
    PRIMARY KEY (UID, RID)  
) ;

**Functional Dependencies**

RID, UID -> Quantity

**Sample Data**

<table>
<thead>
<tr>
<th>rid</th>
<th>uid</th>
<th>quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>R01</td>
<td>U01</td>
<td>1</td>
</tr>
<tr>
<td>R02</td>
<td>U02</td>
<td>2</td>
</tr>
<tr>
<td>R03</td>
<td>U03</td>
<td>1</td>
</tr>
<tr>
<td>R04</td>
<td>U04</td>
<td>1</td>
</tr>
<tr>
<td>R04</td>
<td>U05</td>
<td>1</td>
</tr>
<tr>
<td>R04</td>
<td>U06</td>
<td>1</td>
</tr>
<tr>
<td>R05</td>
<td>U07</td>
<td>1</td>
</tr>
<tr>
<td>R06</td>
<td>U08</td>
<td>1</td>
</tr>
<tr>
<td>R06</td>
<td>U09</td>
<td>1</td>
</tr>
<tr>
<td>R06</td>
<td>U10</td>
<td>1</td>
</tr>
<tr>
<td>R06</td>
<td>U11</td>
<td>1</td>
</tr>
</tbody>
</table>
**Photos**

**Purpose**

This table is used to store information about photos for the recipes, such as height and width in pixels, the date they were added, and whether or not they have been edited.

**Create Statement**

CREATE TABLE Photos (  
  PhID text not null,  
  HeightPX integer not null,  
  WidthPX integer not null,  
  DateAdded date not null,  
  Edited boolean not null,  
  PRIMARY KEY (PhID)  
);

**Functional Dependencies**

PhID -> HeightPX, WidthPX, DateAdded, Edited

**Sample Data**

<table>
<thead>
<tr>
<th>phid</th>
<th>heightpx</th>
<th>widthpx</th>
<th>dateadded</th>
<th>edited</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>640</td>
<td>800</td>
<td>2016-11-23</td>
<td>t</td>
</tr>
<tr>
<td>2</td>
<td>406</td>
<td>305</td>
<td>2016-11-24</td>
<td>t</td>
</tr>
<tr>
<td>3</td>
<td>406</td>
<td>305</td>
<td>2016-11-24</td>
<td>t</td>
</tr>
<tr>
<td>4</td>
<td>440</td>
<td>400</td>
<td>2016-11-26</td>
<td>t</td>
</tr>
<tr>
<td>5</td>
<td>206</td>
<td>605</td>
<td>2016-11-30</td>
<td>t</td>
</tr>
<tr>
<td>6</td>
<td>606</td>
<td>405</td>
<td>2016-11-24</td>
<td>t</td>
</tr>
<tr>
<td>7</td>
<td>606</td>
<td>405</td>
<td>2016-11-24</td>
<td>t</td>
</tr>
<tr>
<td>8</td>
<td>206</td>
<td>315</td>
<td>2016-11-24</td>
<td>t</td>
</tr>
<tr>
<td>9</td>
<td>706</td>
<td>915</td>
<td>2016-11-24</td>
<td>t</td>
</tr>
<tr>
<td>10</td>
<td>406</td>
<td>315</td>
<td>2016-11-24</td>
<td>t</td>
</tr>
<tr>
<td>11</td>
<td>406</td>
<td>315</td>
<td>2016-11-24</td>
<td>t</td>
</tr>
<tr>
<td>12</td>
<td>404</td>
<td>315</td>
<td>2016-11-24</td>
<td>t</td>
</tr>
</tbody>
</table>
**PhotosList**

**Purpose**

This table is used to store information about which photos are associated with a particular recipe.

**Create Statement**

```
CREATE TABLE PhotosList (  
    PhID text not null references Photos(PhID),  
    RID text not null references Recipes(RID),  
    PRIMARY KEY (PhID, RID)  
);  
```

**Functional Dependencies**

PhID, RID ->

**Sample Data**

<table>
<thead>
<tr>
<th>phid</th>
<th>rid</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH01</td>
<td>R01</td>
</tr>
<tr>
<td>PH02</td>
<td>R02</td>
</tr>
<tr>
<td>PH03</td>
<td>R03</td>
</tr>
<tr>
<td>PH04</td>
<td>R03</td>
</tr>
<tr>
<td>PH05</td>
<td>R04</td>
</tr>
<tr>
<td>PH06</td>
<td>R05</td>
</tr>
<tr>
<td>PH07</td>
<td>R06</td>
</tr>
<tr>
<td>PH08</td>
<td>R07</td>
</tr>
<tr>
<td>PH09</td>
<td>R07</td>
</tr>
<tr>
<td>PH10</td>
<td>R08</td>
</tr>
<tr>
<td>PH11</td>
<td>R09</td>
</tr>
<tr>
<td>PH12</td>
<td>R10</td>
</tr>
</tbody>
</table>
**Steps**

**Purpose**

This table is used to store the information for all the steps for each recipe, such as the recipe each are associated with, its step number, and the text that makes up the instructions for the step.

**Create Statement**

```sql
CREATE TABLE Steps (  
  StepNum integer not null,  
  RID text not null references Recipes(RID),  
  Descr text not null,  
  PRIMARY KEY (StepNum, RID)  
);
```

**Functional Dependencies**

StepNum, RID -> Descr

**Sample Data**

<table>
<thead>
<tr>
<th>stepnum</th>
<th>rid</th>
<th>descr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R01</td>
<td>1. Preheat the oven to 425 degrees Fahrenheit and line a rimmed baking sheet with parchment paper. Place the butternut squash on the pan and...</td>
</tr>
<tr>
<td>2</td>
<td>R01</td>
<td>2. Turn the squash face down and roast until it is tender and completely cooked through, about 45 to 50 minutes. Set the squash aside until...</td>
</tr>
<tr>
<td>3</td>
<td>R01</td>
<td>3. Meanwhile, in a medium skillet (or large soup pot, if you'll be serving soup from that pot), warm 1 tablespoon olive oil over medium heat...</td>
</tr>
<tr>
<td>4</td>
<td>R01</td>
<td>4. If you have a high performance blender like a Vitamix (see notes if you are using an immersion blender instead), transfer the cooked shall...</td>
</tr>
<tr>
<td>5</td>
<td>R01</td>
<td>5. If you would like to thin out your soup a bit more, add the remaining cup of broth (I used the full 4 cups, but if you used a small squa...</td>
</tr>
<tr>
<td>6</td>
<td>R01</td>
<td>6. Serve immediately (I like to top each bowl with a little more black pepper). Let leftover soup cool completely before transferring it to...</td>
</tr>
<tr>
<td>7</td>
<td>R02</td>
<td>1. Preheat the oven to 350 degrees F. Grease and flour 2 (8 1/2 by 4 1/4 by 2 1/2-inch) loaf pans. You may also line the bottom with parchme...</td>
</tr>
<tr>
<td>8</td>
<td>R02</td>
<td>2. Cream the butter and 2 cups granulated sugar in the bowl of an electric mixer fitted with the paddle attachment, until light and fluffy, ...</td>
</tr>
<tr>
<td>9</td>
<td>R02</td>
<td>3. Sift together the flour, baking powder, baking soda, and salt in a bowl. In another bowl, combine 1/4 cup lemon juice, the buttermilk, ar...</td>
</tr>
<tr>
<td>10</td>
<td>R02</td>
<td>4. Combine 1/2 cup granulated sugar with 1/2 cup lemon juice in a small saucepan and cook over low heat until the sugar dissolves. When the...</td>
</tr>
</tbody>
</table>
Ingredients

Purpose

This table is used to store information about the ingredients, such as their name, the season that they are best, and whether or not they are an allergen.

Create Statement

CREATE TABLE Ingredients (  
   IID text not null,  
   Name text not null,  
   Season text not null,  
   Allergen boolean,  
   PRIMARY KEY (IID)  
);

Functional Dependencies

IID -> Name, Season, Allergen

Sample Data

<table>
<thead>
<tr>
<th>iid</th>
<th>name</th>
<th>season</th>
<th>allergen</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Butternut Squash</td>
<td>Fall</td>
<td>f</td>
</tr>
<tr>
<td>2</td>
<td>Olive Oil</td>
<td>n/a</td>
<td>f</td>
</tr>
<tr>
<td>3</td>
<td>Shallot</td>
<td>Fall</td>
<td>f</td>
</tr>
<tr>
<td>4</td>
<td>Salt</td>
<td>n/a</td>
<td>f</td>
</tr>
<tr>
<td>5</td>
<td>Garlic</td>
<td>Summer</td>
<td>f</td>
</tr>
<tr>
<td>6</td>
<td>Maple Syrup</td>
<td>Winter</td>
<td>f</td>
</tr>
<tr>
<td>7</td>
<td>Nutmeg</td>
<td>Fall</td>
<td>f</td>
</tr>
<tr>
<td>8</td>
<td>Black Pepper</td>
<td>n/a</td>
<td>f</td>
</tr>
<tr>
<td>9</td>
<td>Vegetable Broth</td>
<td>n/a</td>
<td>f</td>
</tr>
<tr>
<td>10</td>
<td>Butter</td>
<td>n/a</td>
<td>f</td>
</tr>
<tr>
<td>11</td>
<td>Anchovies</td>
<td>n/a</td>
<td>f</td>
</tr>
<tr>
<td>12</td>
<td>Rosemary</td>
<td>Fall/Winter</td>
<td>f</td>
</tr>
<tr>
<td>13</td>
<td>Apples</td>
<td>Fall</td>
<td>f</td>
</tr>
<tr>
<td>14</td>
<td>Caramel</td>
<td>n/a</td>
<td>f</td>
</tr>
</tbody>
</table>
**IngredientsList**

**Purpose**

This table is used to store the information about which ingredients go with each part of a recipe, the way they should be prepared, and the quantity needed. This table basically represents the ingredients list that is typically seen on recipes.

**Create Statement**

```sql
CREATE TABLE IngredientsList (  
    IID text not null references Ingredients(IID),
    ParID text not null references Parts(ParID),
    WayPrep text not null,
    Quantity text not null,
    PRIMARY KEY (IID, ParID, WayPrep)
);  
```

**Functional Dependencies**

IID, ParID, WayPrep --> Quantity

**Sample Data**

<table>
<thead>
<tr>
<th>iid</th>
<th>parid</th>
<th>wayprep</th>
<th>quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I01</td>
<td>Chop in half and seed</td>
<td>1 Large</td>
</tr>
<tr>
<td>2</td>
<td>I02</td>
<td>None</td>
<td>1 Tablespoon</td>
</tr>
<tr>
<td>3</td>
<td>I03</td>
<td>Chopped</td>
<td>1/2 cup</td>
</tr>
<tr>
<td>4</td>
<td>I04</td>
<td>None</td>
<td>1 Teaspoon</td>
</tr>
<tr>
<td>5</td>
<td>I05</td>
<td>Pressed or Minced</td>
<td>4 or 5</td>
</tr>
<tr>
<td>6</td>
<td>I06</td>
<td>None</td>
<td>1 Teaspoon</td>
</tr>
<tr>
<td>7</td>
<td>I07</td>
<td>None</td>
<td>1/8 Teaspoon</td>
</tr>
<tr>
<td>8</td>
<td>I08</td>
<td>Grind</td>
<td>To Taste</td>
</tr>
<tr>
<td>9</td>
<td>I09</td>
<td>None</td>
<td>Vegetable Broth</td>
</tr>
<tr>
<td>10</td>
<td>I10</td>
<td>None</td>
<td>1 or 2 Tablespoons</td>
</tr>
<tr>
<td>11</td>
<td>I11</td>
<td>Divided</td>
<td>2 1/2 Cups</td>
</tr>
<tr>
<td>12</td>
<td>I12</td>
<td>at room temperature.</td>
<td>4 Large</td>
</tr>
<tr>
<td>13</td>
<td>I13</td>
<td>Zest</td>
<td>1/3 cup</td>
</tr>
<tr>
<td>14</td>
<td>I13</td>
<td>Juice</td>
<td>3/4 Cup</td>
</tr>
<tr>
<td>15</td>
<td>I14</td>
<td>None</td>
<td>3 Cups</td>
</tr>
</tbody>
</table>
**Parts**

**Purpose**

This table is used to store the important information associated with each part of a recipe. When looking at the ingredients and instructions for a recipe, it is critical to know the part that they are associated so that the correct quantities of each ingredient and the proper techniques or preparatory methods can be used.

**Create Statement**

```sql
CREATE TABLE Parts (  
  ParID text not null,  
  Name text not null,  
  PRIMARY KEY (ParID)  
);
```

**Functional Dependencies**

ParID -> Name

**Sample Data**

<table>
<thead>
<tr>
<th>ParID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PAR01 Soup</td>
</tr>
<tr>
<td>2</td>
<td>PAR02 Lemon Cake</td>
</tr>
<tr>
<td>3</td>
<td>PAR03 Lemon Glaze</td>
</tr>
<tr>
<td>4</td>
<td>PAR04 Pulled Pork</td>
</tr>
<tr>
<td>5</td>
<td>PAR05 Taco Assembly</td>
</tr>
<tr>
<td>6</td>
<td>PAR06 Entire Recipe</td>
</tr>
<tr>
<td>7</td>
<td>PAR07 The Chicken Mixture</td>
</tr>
<tr>
<td>8</td>
<td>PAR08 Finishing the dish</td>
</tr>
<tr>
<td>9</td>
<td>PAR09 Entire Recipe</td>
</tr>
<tr>
<td>10</td>
<td>PAR10 Entire Recipe</td>
</tr>
<tr>
<td>11</td>
<td>PAR11 Entire Recipe</td>
</tr>
<tr>
<td>12</td>
<td>PAR12 Entire Recipe</td>
</tr>
<tr>
<td>13</td>
<td>PAR13 Dough</td>
</tr>
<tr>
<td>14</td>
<td>PAR14 Coffee Cream</td>
</tr>
</tbody>
</table>
PartList

Purpose

This table is used to store information about which parts go with each particular recipe. This is critical because a recipe is not complete without all of its parts, and the ingredients associated with each part must come together as a single ingredients list for the recipe.

Create Statement

CREATE TABLE PartList (  
  ParID text not null references Parts(ParID),  
  RID text not null references Recipes(RID),  
  PRIMARY KEY (ParID, RID)  
);

Functional Dependencies

ParID, RID ->

Sample Data

<table>
<thead>
<tr>
<th>parid</th>
<th>rid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R01</td>
</tr>
<tr>
<td>2</td>
<td>R02</td>
</tr>
<tr>
<td>3</td>
<td>R02</td>
</tr>
<tr>
<td>4</td>
<td>R03</td>
</tr>
<tr>
<td>5</td>
<td>R03</td>
</tr>
<tr>
<td>6</td>
<td>R04</td>
</tr>
<tr>
<td>7</td>
<td>R05</td>
</tr>
<tr>
<td>8</td>
<td>R05</td>
</tr>
<tr>
<td>9</td>
<td>R06</td>
</tr>
<tr>
<td>10</td>
<td>R07</td>
</tr>
<tr>
<td>11</td>
<td>R08</td>
</tr>
<tr>
<td>12</td>
<td>R09</td>
</tr>
<tr>
<td>13</td>
<td>R10</td>
</tr>
<tr>
<td>14</td>
<td>R10</td>
</tr>
</tbody>
</table>
Substitutions

Purpose

This table is used to store information about ingredient substitutions, such as the ingredient that is being substituted for, how much the substitution makes, and the instructions for doing the substitution. This information allows the chef to successfully replace one ingredient for another if they do not have the original ingredient called for.

Create Statement

CREATE TABLE Substitutions (  
    SLID text not null,  
    IID text not null references Ingredients(IID),  
    HowMuchMake text not null,  
    Instructions text not null,  
    PRIMARY KEY (SLID)  
);  

Functional Dependencies

SLID -> IID, HowMuchMake, Instructions

Sample Data
IngredientsOfSubstitution

Purpose

This table is used to store information about which ingredients are needed to complete a particular substitution, and the quantity of each.

Create Statement

CREATE TABLE IngredientsOfSubstitution (
  SLID text not null references Substitutions(SLID),
  IID text not null references Ingredients(IID),
  Quantity text not null,
  PRIMARY KEY (SLID, IID)
);

Functional Dependencies

SLID, IID -> Quantity

Sample Data

<table>
<thead>
<tr>
<th>slid</th>
<th>iid</th>
<th>quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL01</td>
<td>I58</td>
<td>About 1 Cup</td>
</tr>
<tr>
<td>SL01</td>
<td>I13</td>
<td>1 Tablespoon</td>
</tr>
<tr>
<td>SL02</td>
<td>I37</td>
<td>As much as the recipe calls for.</td>
</tr>
<tr>
<td>SL03</td>
<td>I42</td>
<td>As much as the recipe calls for.</td>
</tr>
<tr>
<td>SL04</td>
<td>I61</td>
<td>As much as the recipe calls for.</td>
</tr>
</tbody>
</table>
FlavorAffinities

Purpose

This table is used to store information the Flavor Affinities, such as their origin. For example, a flavor affinity may be of Mexican origin, meaning that the flavor combos it creates reflect Mexican cuisine. This is useful for chefs, as it tells them which particular flavors to combine to imitate a particular cuisine.

Create Statement

CREATE TABLE FlavorAffinities (  
  AID text not null,  
  Origin text,  
  PRIMARY KEY (AID)  
);

Functional Dependencies

AID -> Origin

Sample Data

<table>
<thead>
<tr>
<th>aid text</th>
<th>origin text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A01 African Cuisine West</td>
</tr>
<tr>
<td>2</td>
<td>A02 Spanish Cuisine</td>
</tr>
<tr>
<td>3</td>
<td>A03 American Cuisine</td>
</tr>
<tr>
<td>4</td>
<td>A04 Asian Cuisine</td>
</tr>
<tr>
<td>5</td>
<td>A05 French Cuisine</td>
</tr>
<tr>
<td>6</td>
<td>A06 Mexican Cuisine</td>
</tr>
<tr>
<td>7</td>
<td>A07 Mexican Cuisine</td>
</tr>
</tbody>
</table>
**AffinitiesList**

**Purpose**

This table is used to store the association of particular ingredients with a particular flavor affinity. All the ingredients that are members of the affinity can be combined by the chef to make their dish resemble the cuisine of origin, and to create harmonious flavors.

**Create Statement**

CREATE TABLE AffinitiesList (  
   AID     text not null references FlavorAffinities(AID),  
   IID     text not null references Ingredients(IID),  
   PRIMARY KEY (AID, IID)  
);

**Functional Dependencies**

AID, IID ->

**Sample Data**

<table>
<thead>
<tr>
<th></th>
<th>aid</th>
<th>lid</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A01</td>
<td>I43</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>A01</td>
<td>I29</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>A01</td>
<td>I48</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>A02</td>
<td>I49</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>A02</td>
<td>I50</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>A02</td>
<td>I02</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>A02</td>
<td>I13</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>A03</td>
<td>I51</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>A03</td>
<td>I52</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>A03</td>
<td>I48</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>A04</td>
<td>I53</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>A04</td>
<td>I10</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>A04</td>
<td>I25</td>
<td></td>
</tr>
</tbody>
</table>
Themes

Purpose

This table allows us to identify different themes that could be associated with a publication method or recipe in particular. This allows us to categorize the recipes and publications based on their themes. It also gives us information about a particular theme, such as its name and description.

Create Statement

CREATE TABLE Themes (  
  TID text not null,  
  Name text not null,  
  Descr text,  
  PRIMARY KEY (TID)  
);

Functional Dependencies

TID -> Name, Descr

Sample Data

<table>
<thead>
<tr>
<th>tid</th>
<th>name</th>
<th>descr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T01</td>
<td>Comfort Food you want to eat on a cold day with family.</td>
</tr>
<tr>
<td>2</td>
<td>T02</td>
<td>Vegetarian To cook without the use of any meat.</td>
</tr>
<tr>
<td>3</td>
<td>T03</td>
<td>Dessert A nice way to finish a meal.</td>
</tr>
<tr>
<td>4</td>
<td>T04</td>
<td>Simple Good for mid-week cooking/baking.</td>
</tr>
<tr>
<td>5</td>
<td>T05</td>
<td>Mexican Spicy and sweet.</td>
</tr>
<tr>
<td>6</td>
<td>T06</td>
<td>Chipolte Spicy.</td>
</tr>
<tr>
<td>7</td>
<td>T07</td>
<td>Indulgence Foods you love but don't eat that often.</td>
</tr>
<tr>
<td>8</td>
<td>T08</td>
<td>Classics The basic recipes.</td>
</tr>
<tr>
<td>9</td>
<td>T09</td>
<td>In the Family Passed down recipes.</td>
</tr>
<tr>
<td>10</td>
<td>T10</td>
<td>Italian Dessert Classic to the Italian cuisine and a good way to end a meal.</td>
</tr>
</tbody>
</table>
RecipeThemesList

Purpose

This table allows us to identify the particular themes that are associated with a particular recipe.

Create Statement

CREATE TABLE RecipeThemesList (  
  TID text not null references Themes(TID),  
  RID text not null references Recipes(RID),  
  PRIMARY KEY (TID, RID)  
);

Functional Dependencies

TID, RID ->

Sample Data

<table>
<thead>
<tr>
<th></th>
<th>tid</th>
<th>rid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T01</td>
<td>R01</td>
</tr>
<tr>
<td>2</td>
<td>T03</td>
<td>R02</td>
</tr>
<tr>
<td>3</td>
<td>T05</td>
<td>R03</td>
</tr>
<tr>
<td>4</td>
<td>T02</td>
<td>R04</td>
</tr>
<tr>
<td>5</td>
<td>T06</td>
<td>R05</td>
</tr>
<tr>
<td>6</td>
<td>T07</td>
<td>R06</td>
</tr>
<tr>
<td>7</td>
<td>T01</td>
<td>R06</td>
</tr>
<tr>
<td>8</td>
<td>T04</td>
<td>R06</td>
</tr>
<tr>
<td>9</td>
<td>T08</td>
<td>R07</td>
</tr>
<tr>
<td>10</td>
<td>T01</td>
<td>R07</td>
</tr>
<tr>
<td>11</td>
<td>T04</td>
<td>R07</td>
</tr>
</tbody>
</table>
**PublishMethodsThemesList**

**Purpose**

This table is used to store the association of themes with each publication. Thus, it allows us to describe the style of the publication using particular themes.

**Create Statement**

CREATE TABLE PublishMethodsThemesList (  
    TID text not null references Themes(TID),  
    PID text not null references PublishMethods(PID),  
    PRIMARY KEY (TID, PID)  
);  

**Functional Dependencies**

TID, PID ->

**Sample Data**

<table>
<thead>
<tr>
<th>tid</th>
<th>pid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T02</td>
</tr>
<tr>
<td>2</td>
<td>T04</td>
</tr>
<tr>
<td>3</td>
<td>T04</td>
</tr>
<tr>
<td>4</td>
<td>T02</td>
</tr>
<tr>
<td>5</td>
<td>T05</td>
</tr>
<tr>
<td>6</td>
<td>T07</td>
</tr>
<tr>
<td>7</td>
<td>T08</td>
</tr>
<tr>
<td>8</td>
<td>T08</td>
</tr>
<tr>
<td>9</td>
<td>T04</td>
</tr>
<tr>
<td>10</td>
<td>T04</td>
</tr>
</tbody>
</table>
Views

EntireRecipe

Purpose

This view shows the entire recipe with all of its associated information. Essentially, this view shows all the information you need to access the different parts of a recipe. Using this view, you can get the ingredients, recipe information, utensils, photos, steps, and parts associated with this recipe. For a chef this would be necessary so that they can make the recipe.

Create Statement

cREATE VIEW EntireRecipe AS
SELECT  r.name, 
       r.datecreated, 
       r.preptimemin, 
       r.cooktimemin, 
       r.serves, 
       r.src, 
       ul.uid, 
       pl.ParID, 
       rg.pid, 
       phl.phid, 
       s.descr
FROM    Recipes r 
        INNER JOIN utensilslist ul ON r.rid = ul.rid 
        INNER JOIN PartsList pl ON r.rid = pl.rid 
        INNER JOIN recipegroups rg ON r.rid = rg.rid 
        INNER JOIN photoslist phl ON r.rid = phl.rid 
        INNER JOIN Steps s ON r.rid = s.rid 
        INNER JOIN RecipeThemesList rtl ON r.rid = rtl.rid;
**IngredientAffinities**

**Purpose**

This view shows information about all the ingredients involved in each flavor affinity as well as the information about the affinity itself. This would be necessary for a chef who wanted to see which ingredients combine well to make a particular flavor palette.

**Create Statement**

```sql
create view IngredientAffinities AS
Select  al.aid,
       i.name,
       i.allergen,
       fa.origin
from Affinitieslist al inner join ingredients i ON i.iid = al.iid
inner join flavoraffinities fa ON fa.aid = al.aid;
```
**Publication Information**

**Purpose**

This view shows comprehensive information about the publications. It expands the general publication information to include the themes associated with the publication and its authors. This would be necessary for a chef who wished to learn more about a particular source or someone who wanted to find the publication.

**Create Statement**

```sql
CREATE VIEW publicationinformation AS
SELECT p.pid,
   p.title,
   p.yearpublished,
   p.contentdescr,
   peo.perid,
   peo.fname,
   peo.lname,
   t.tid,
   t.name
FROM Published pd,
     publishmethods p,
     authors a,
     people peo,
     Publishmethodsthemelist ptl,
     Themes t
WHERE pd.pid = p.pid
  AND pd.perid = a.perid
  AND peo.perid = a.perid
  AND ptl.pid = p.pid
  AND ptl.tid = t.tid;
```
Reports

Number of Recipes Published by an Author in a Particular Year

Food Network is a company that hires many different chefs and authors. This report can be used to check the productivity of their chefs and authors to ensure that they are keeping an appropriate rate of publication by showing the number of publications that they authored in a particular year. The year that the person desires is indicated by YEAR.

Query

```
select peo.fname, peo.lname, count(*) as NumberOfPublications
from publishmethods p,
    published pu,
    authors a,
    people peo
where peo.perid = a.perid
    and p.pid = pu.pid
    and pu.perid = a.perid
    and yearpublished = YEAR
group by peo.fname, peo.lname;
```

This report will work with user input to generate the appropriate report based on the year that is entered. In the query above, I have written YEAR where the user-entered year would go. For example, if the user wanted a report about 2016, then the query would look as follows.

Query:

```
select peo.fname, peo.lname,
    count(*) as NumberOfPublications
from publishmethods p,
    published pu,
    authors a,
    people peo
where peo.perid = a.perid
    and p.pid = pu.pid
    and pu.perid = a.perid
    and yearpublished = '2016'
group by peo.fname,
    peo.lname;
```
Average Complexity of Recipes in the Database

As a company, Food Network is concerned with the average complexity of the recipes that they are posting to ensure that they have a good balance of complex and simple recipes for their viewers to choose from. The complexity of a recipe is determined based on the number of utensils it uses, the number of parts and steps it has. This report shows the average number of utensils used, and the average number of parts and steps associated with recipes. If the average of any one of these things is too high, then Food Network knows it must add recipes that are simpler in that particular area to lower the average.

Query

```
select *
from (select avg(num) as averageUtensils
from (Select count(ul.uid) as num
    from recipes r, utensilslist ul, utensils u
    where r.rid = ul.rid
    and ul.uid = u.uid
    group by r.rid
    ) as utensilcount
    ) as utensils,
    (select avg(num1) as averageSteps
from (Select count(steps.stepnum) as num1
    from recipes r, steps
    where r.rid = steps.rid
    group by r.rid
    ) as stepscount
    ) as average2,
    (select avg(num2) as averageParts
from (Select count(pl.parid) as num2
    from recipes r, parts p, partslist pl
    where r.rid = pl.rid
    and pl.parid = p.parid
    group by r.rid
    ) as partscount
    ) as average3;
```
Theme Usage

This report shows how many times each theme is being used. This data report can be used to figure out which themes are being underused and need to be developed further. Unfortunately, it will not show themes that are not used by both recipes and publications, so it is up to the admin to know a list of the themes they have.

Query

```sql
select COALESCE(recipetheme.rtheme,'Not Used') AS RThemes,
       COALESCE(recipetheme.ruse,'0') AS RNumTimesUsed,
       COALESCE(publishtheme.ptHEME,'Not Used') AS PThemes,
       COALESCE(publishtheme.puse,'0') AS PNumTimesUsed
from (select count(rtl.rid) as ruse,
           rtl.tid as rtheme
      from recipes r,
           recipethemeslist rtl
      where r.rid = rtl.rid
      group by rtl.tid
     ) as recipetheme full outer join
     (select count(ptl.tid) as puse,
          ptl.tid as ptheme
     from publishmethodsthemelist ptl,
         publishmethods pm
     where pm.pid = ptl.pid
     group by ptl.tid
    ) as publishtheme ON recipetheme.rtheme =
publishtheme.ptheme
Order by rthemes;
```
Potential Tweaks to a Recipe

Food network is always looking to tweak their recipes in new and interesting ways. This report shows which cuisine the ingredients in the recipe are associated with. By showing this, it tells the chefs which flavor palettes of different cuisines to draw from to enhance the recipe. As well, it also shows the flavor affinities associated with the ingredients in each recipe. This can be used to figure out which ingredients to add to the recipe to change the flavor in a way that is pleasant.

Query

```sql
select  r.rid,
        r.name,
        fa.origin,
        al.*,
        i.name
from    recipes r,
        partslist pl,
        ingredientslist il,
        ingredients i,
        flavoraffinities fa,
        affinitieslist al
where   r.rid = pl.rid
        and pl.parid = il.parid
        and il.iid = i.iid
        and fa.aid = al.aid
        and i.iid = al.iid
order by r.rid
```
Stored Procedures

Get_Recipe_Ingredientslist_byName

Purpose:

This method works with the method get_recipe_ingredientslist_bynameORID to allow the retrieval of the recipe ingredients list by Name.

Query:

create or replace function get_recipe_ingredientslist_byName(text)
returns table (iid text,
     recipename text,  
     parid text,      
     wayprep text,    
     quantity text)
as
$$
declare
    recipeName text := $1;
begin
    return query
    select ingredients.iid, 
        ingredients.name, 
        ingredientslist.parid, 
        ingredientslist.wayprep, 
        ingredientslist.quantity 
    from Ingredients, 
        Ingredientslist, 
        partslst, 
        recipes 
    where Ingredients.iid = Ingredientslist.iid 
    and partslst.parid = ingredientslist.parid 
    and partslst.rid = recipes.rid 
    and recipes.name = recipeName;
end;
$$
language plpgsql;
Get_Recipe_Ingredientslist_byID

**Purpose:**

This method allows database user to retrieve all ingredients associated with a recipe by entering the recipe ID.

**Create Statement:**

```sql
create or replace function get_recipe_ingredientslist_byID(text)
returns table (iid text,
    recipename text,
    parid text,
    wayprep text,
    quantity text)
as
$$
declare
    recipeID text := $1;
begin
    return query
        select ingredients.iid,
            ingredients.name,
            ingredientslist.parid,
            ingredientslist.wayprep,
            ingredientslist.quantity
        from Ingredients,
            Ingredientslist,
            partslist
        where Ingredients.iid = Ingredientslist.iid
            and partslist.parid = ingredientslist.parid
            and partslist.rid = recipeID;
end;
$$
language plpgsql;
```
Get_Recipe_Ingredientslist_byNameORID

Purpose:

The purpose of this function is to facilitate the use of the database. It allows users to either enter the ID or the name of a recipe that they want to find the ingredients list for and it will return the correct list of ingredients from the database.

Create Statement:

```
create or replace function get_recipe_ingredientslist_byNameOrId(text, text, REFCURSOR) returns refcursor as $$
declare
  recipeName text  := $1;
  recipeId  text   := $2;
  resultset REFCURSOR := $3;
begin
  if (recipeId IS NOT NULL) then
    open resultset for
    select *
    FROM get_recipe_ingredientslist_byId(recipeId);
  else
    open resultset for
    select *
    FROM get_recipe_ingredientslist_byName(recipeName);
  end if;

  return resultset;
end;
$$
language plpgsql;
```
**calculateTotalTimeMin**

**Purpose**

This function is used to calculate or recalculate the total time for a recipe each time a recipe record is inserted or updated.

**Query**

create or replace function calculateTotalTimeMin() returns trigger as
$$
\text{declare}
\text{total integer := cast(new.preptimemin as Integer) + cast(new.cooktimemin as integer);} 
\text{begin}
\text{new.totaltimemin = total;}
\text{return NEW;}
\text{end;}
$$
language plpgsql;

**Triggers**

**TotalTimeCheck**

**Purpose**

This trigger is called after a recipe record is updated or inserted into the recipes table. It calls the stored procedure calculateTotalTimeMin() to ensure that the total time is correct.

**Query**

create trigger totalTime
BEFORE INSERT OR UPDATE ON Recipes
FOR EACH ROW EXECUTE PROCEDURE calculateTotalTimeMin();
Security

There are 6 primary users of this database: chefs, authors, administrators, photographers, affinity experts, and theme writers. For each role, the user is revoked of all privileges before being granted the appropriate privileges. To reduce the length of this section, the revoke statements have been excluded.

Administrators

Administrators are the employees at Food Network that manage the recipes database. They must be able to edit, select or delete any information that they desire in order to maintain the accuracy of the data and remove unneeded information. They are also in charge of matching recipe and publications with their appropriate themes.

GRANT SELECT, INSERT, UPDATE, DELETE ON entirerecipe TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Recipes TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON steps TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Ingredients TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON IngredientsList TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON PhotosList TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Photos TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Parts TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON PartsList TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON UtensilsList TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Utensils TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON AffinitiesList TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON FlavorAffinities TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON ingredientaffinities TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON RecipeThemesList TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON PublishMethodsThemesList TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Themes TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON IngredientsOfSubstitution TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Substitutions TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Published TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE on publicationinformation TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Published TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON People TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Chefs TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON CookbookAuthors TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON BlogAuthors TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON PublishMethods TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Blogs TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON Cookbooks TO admins;
GRANT SELECT, INSERT, UPDATE, DELETE ON RecipeGroups TO admins;

Chefs

Chefs need the privileges to access any information about recipes in the database so that they can make them and use them as inspiration for new recipes.

GRANT SELECT ON EntireRecipe TO chefs;
GRANT SELECT ON recipes TO chefs;
GRANT SELECT ON Parts TO chefs;
GRANT SELECT ON Partslist TO chefs;
GRANT SELECT ON Ingredientslist TO chefs;
GRANT SELECT ON Ingredients TO chefs;
GRANT SELECT ON IngredientsOfSubstitution TO chefs;
GRANT SELECT ON Substitutions TO chefs;
GRANT SELECT ON FlavorAffinities TO chefs;
GRANT SELECT ON AffinitiesList TO chefs;
GRANT SELECT ON Steps TO chefs;
GRANT SELECT ON IngredientAffinities TO chefs;
GRANT SELECT ON publicationinformation TO chefs;

Authors

This group of users needs to be able to see the information about the recipes that have been published, the other authors and all of the different publications. They are able to insert new publications, however they are not allowed to update or delete existing records.

GRANT SELECT, INSERT ON entirerecipe TO authors;
GRANT SELECT, INSERT ON publicationinformation TO authors;
GRANT SELECT, INSERT ON PublishMethods TO authors;
GRANT SELECT, INSERT ON Blogs TO authors;
GRANT SELECT, INSERT ON Cookbooks TO authors;
GRANT SELECT, INSERT ON RecipeGroups TO authors;
GRANT SELECT, INSERT ON steps TO authors;
GRANT SELECT, INSERT ON IngredientsList TO authors;
GRANT SELECT, INSERT ON Ingredients TO authors;
GRANT SELECT, INSERT ON PartsList TO authors;
GRANT SELECT, INSERT ON Parts TO authors;
GRANT SELECT, INSERT ON UtensilsList TO authors;
GRANT SELECT, INSERT ON Utensils TO authors;
GRANT SELECT, INSERT ON Recipes TO authors;
GRANT SELECT ON IngredientsOfSubstitution TO authors;
GRANT SELECT ON Substitutions TO authors;
Photographers

This group needs to be able to see the recipes associated with each publication and add photographs to be associated with the different recipes.

GRANT SELECT, INSERT, UPDATE, DELETE ON Photos TO photographers;
GRANT SELECT, INSERT, UPDATE, DELETE ON PhotosList TO photographers;
GRANT SELECT ON EntireRecipe TO photographers;
GRANT SELECT ON RecipeGroups TO photographers;
GRANT SELECT ON PublishMethods TO photographers;
GRANT SELECT ON Cookbooks TO photographers;
GRANT SELECT ON Blogs TO photographers;

Theme Writers

These users need to be able to update, insert and delete themes and their associations with particular recipes as well as access all information about recipes and their publications.

GRANT SELECT, INSERT, UPDATE, DELETE ON Themes TO themeWriters;
GRANT SELECT, INSERT, UPDATE, DELETE ON PublishMethodsThemesList TO themeWriters;
GRANT SELECT, INSERT, UPDATE, DELETE ON RecipeThemesList TO themeWriters;
GRANT SELECT ON publicationinformation TO themeWriters;
GRANT SELECT ON PublishMethods TO themeWriters;
GRANT SELECT ON Cookbooks TO themeWriters;
GRANT SELECT ON Blogs TO themeWriters;
GRANT SELECT ON Recipes TO themeWriters;
GRANT SELECT ON RecipeGroups TO themeWriters;

Affinity Experts

This group of users needs to be able to access the information about all the ingredients as well as modify the affinities and the list of ingredients associated with them.

GRANT SELECT, INSERT, UPDATE, DELETE ON ingredientaffinities TO affinityexperts;
GRANT SELECT, INSERT, UPDATE, DELETE ON AffinitiesList TO affinityexperts;
GRANT SELECT, INSERT, UPDATE, DELETE ON FlavorAffinities TO affinityexperts;
GRANT SELECT, INSERT ON Ingredients TO affinityexperts;
Implementation Notes

The following are suggestions for implementation:

1. When a recipe is entered into the database, it should be entered with all of its corresponding data at once in order to ensure that the complete recipe is present.
2. A large number of flavor affinities from different style cuisines should be entered for the first use of the database so that the ingredients of recipes looking for affinities can have a large repository to draw from.
3. If an author realizes that information about a publication they are associated with is incorrect, they should notify an admin to update the information accordingly.

Known Problems

The following are known problems with the database:

1. An author could in theory add false publications to the database that do not actually exist. The accuracy of the publications records are thus dependent on the honestly of the authors and the diligence of the admins in checking the database. The only prevention that is built in against this is that authors cannot add authors and or link them to particular publications. This means that for any authors to be associated with a publication, the publication must be reviewed by the admins first.
2. The reference for an alternate utensil is a text name. To make the database as useful as possible, this should actually be a Utensil ID so that the user can get the needed information about the alternate utensil.
3. The number of cookbooks published by an author is stored in the database, however this number will change as the database grows and the authors add new publications. Thus, this is definitely a source for inaccuracy and should be addressed in some way by the administration.
4. Additionally, there is no way to ensure that a recipe is associated with the correct publication method, meaning the administrators and the authors must check this.
5. Finally, there is no way to ensure that a publication is entered in only the cookbooks subtype or the blogs subtype. Obviously a publication can only be one at a time, so the users of the database must watch this.

Future Enhancements

The following are suggestions for future enhancements that may be desired:

1. First, the database has the potential to grow with the enterprise that it serves. For example, additional methods of publication and types of authors can be added by simply adding an additional table for each.
2. The amount of information that is stored about a recipe can be increased easily, so as the information about recipes that is important changes, the database will change as well.

3. There is a lot of potential for additional stored procedures to enhance the facility of use by the users. For example, you could implement a stored procedure that automatically makes the association of themes with recipes and publish methods based on the keywords in their descriptions.

4. It may be desired to add a table relating to the source of the recipe, such as a reference to the recipe it was inspired by and the author who originated the inspirational recipe.

5. Certainly an enhancement that would be helpful would be the addition of a way to ensure that a publication is only entered in one subtype of publishmethods. This would prevent the known problem above of falsely declaring a publishmethod as both a cookbook and a blog.

6. Finally, it may be practical to implement a method for determining which flavor affinity would be best to use for an addition to a particular recipe. For example, when making a soup it would be incorrect to use a flavor affinity truly meant for baking.