

SSE 691

**Database Design in Software
Engineering**

Project #3

by

Jason Payne

June 24, 2016

TABLE OF CONTENTS

1. Project Status Review	5
1.1 Final Table List Update	5
2. Table Relationships & Characteristics	7
2.1 Defining Table Relationships	7
2.2 Designating Foreign Keys	8
2.3 Relationship Characteristics	9
2.4 Business Rules	10
3. Database Implementation & Tests	12
3.1 Users	12
3.2 User Preferences	15
3.3 Recipes & Ingredients	17
3.4 Cupboard Items	22
3.5 Menus	24
3.6 Meal Plan	27
3.7 Grocery Lists	28
Non-Direct Activity Report	31

TABLE OF FIGURES

Figure 1: Entity Relationship Diagram with refined Foreign Keys	9
Figure 2: Entity Relationship Diagram with defined table relationships	10
Figure 3: Entity Relationship diagram with Measurement Units validation table.....	11

TABLE OF TABLES

Table 1: Updated Final Table List	5
Table 2: Table Relationship Matrix	7
Table 3: Table Relationship Calculations	8

Topics Covered	Topic Examples
Relational Database Design	<ul style="list-style-type: none">• Table Relationships• Table Characteristics• Database Implementation• Database Tests

1. Project Status Review

For this project, progress will continue on the design of the database for the Recipe App as established in the previous project. In the previous project, database tables were identified, fields were refined and associated with their closest related subjects, table keys were designated, and field specifications were created for each field in the database. Picking up at that point, this project will continue to refine the tables while also defining the relationships and characteristics that will be critical for proper behavior of the database. Finally, the database will be realized by implementing the design laid out in the following sections. Once the database is created, tests and their results will be provided as a final validation step.

1.1 Final Table List Update

Upon further analysis of the 'Menus' table, it was revealed that information was commonly duplicated in the Menu Item, Menu Item Type, Price, Menu Item Popularity Rating, and Order Frequency fields. Each time a new menu recipe (Menu Item) is created, all of the data in the aforementioned fields gets repeated. To address these issues, a new table named 'Menu Items' was created as a subset table to Menus as a container for a specific Menu's menu items, or recipes. The new table and its fields were refined and added to the Final Table List (Table 1).

Table 1: Updated Final Table List

Table	Field List
Users	<ul style="list-style-type: none"> • User Name (PK) • E-mail Address (AK) • Home Address • User Phone Number
User Preferences	<ul style="list-style-type: none"> • User Name (PK) • Preference Name • Preference Value
Cupboard Items	<ul style="list-style-type: none"> • User Name (CPK) • Cupboard Item (CPK) • Cupboard Item UoM • Quantity Remaining • Expiration Date
Ingredients	<ul style="list-style-type: none"> • Ingredient ID (PK) • Ingredient Brand • Ingredient Name • Nutritional Value • Ingredient UoM • Shelf Life

Table	Field List
Recipes	<ul style="list-style-type: none"> • Recipe ID (PK) • User Name (CAK) • Recipe Name (CAK) • Recipe Image • Popularity Rating • Preparation Time • Preparation Steps • Cooking Time • Cooking Instructions • Serving Amount • Recipe Notes
Ingredient List Items	<ul style="list-style-type: none"> • Recipe ID (CPK) • Ingredient List Item (CPK) • Ingredient Quantity • Ingredient List Item UoM
Grocery Lists	<ul style="list-style-type: none"> • Grocery List ID (PK) • User Name • Grocery List Name • Grocery List Save Date
Grocery List Items	<ul style="list-style-type: none"> • Grocery List ID (CPK) • Grocery List Item (CPK) • Quantity Desired • Quantity Desired UoM
Meal Plan	<ul style="list-style-type: none"> • Meal Plan ID (PK) • User Name • Date • Meal Type • Meal Recipe
Menus	<ul style="list-style-type: none"> • Menu ID (PK) • User Name • Menu Name
Menu Items	<ul style="list-style-type: none"> • Menu Item ID (PK) • Menu Item • Menu Item Type • Price • Menu Item Popularity Rating • Order Frequency

2. Table Relationships & Characteristics

This section will setup the final design of the database by:

- Defining the relationships between tables (one-to-one, one-to-many, many-to-many)
- Identifying and refining foreign keys
- Defining deletion rules for each relationship
- Identifying the type and degree of participation for each table
- Reviewing business rules

2.1 Defining Table Relationships

To determine the magnitude of the table relationship between two tables, the magnitude (one-to-one, one-to-many, many-to-many) must first be determined from both viewpoints (i.e. from the parent to the child, from the child to the parent). The best way to accomplish this is to setup a matrix that details the magnitude from both perspectives. Table 2 illustrates this for the tables in the Recipe App.

Table 2: Table Relationship Matrix

	Users	User Preferences	Cupboard Items	Ingredients	Recipes	Ingredient List Items	Grocery Lists	Grocery List Items	Meal Plan	Menus	Menu Items
Users		1:N	1:N		1:N		1:N		1:N	1:N	
User Preferences	1:1										
Cupboard Items	1:1			1:1							
Ingredients			1:N			1:N		1:N			
Recipes	1:1					1:N			1:N		1:N
Ingredient List Items				1:1	1:1						
Grocery Lists	1:1							1:N			
Grocery List Items				1:1			1:1				
Meal Plan	1:1				1:1						
Menus	1:1										1:N
Menu Items					1:1					1:1	

With the magnitudes from both perspectives determined, the next step is to apply a simple formula that results in the table relationship that will be used in the final design and implementation of the database. The results of the final relationship determination for the tables in the Recipe App can be seen in Table 3.

Table 3: Table Relationship Calculations

Table 1	Table 2	Formula	Result
Users			
	User Preferences	1:N + 1:1	1:N
	Cupboard Items	1:N + 1:1	1:N
	Recipes	1:N + 1:1	1:N
	Grocery Lists	1:N + 1:1	1:N
	Meal Plan	1:N + 1:1	1:N
	Menus	1:N + 1:1	1:N
Ingredients			
	Ingredient List Items	1:N + 1:1	1:N
	Cupboard Items	1:N + 1:1	1:N
	Grocery List Items	1:N + 1:1	1:N
Recipes			
	Ingredient List Items	1:N + 1:1	1:N
	Meal Plan	1:N + 1:1	1:N
	Menu Items	1:N + 1:1	1:N
Grocery Lists			
	Grocery List Items	1:N + 1:1	1:N
Menus			
	Menu Items	1:N + 1:1	1:N

2.2 Designating Foreign Keys

With the magnitude of table relationships defined, the foreign keys can be identified and refined. The foreign key will be the key that the child table “borrows” from the parent table that establishes the relationship between the two tables. For one-to-one and one-to-many relationships, the primary key becomes the foreign key, so it is common for the foreign key to already exist within a table, but as a different name. To account for this, the foreign keys must be refined in the same manner that all other keys have been established: with a set of criteria known as the Elements of a Foreign Key. The results of this refinement process can be seen in Figure 1. Any changes to pre-existing field names have been highlighted in yellow.

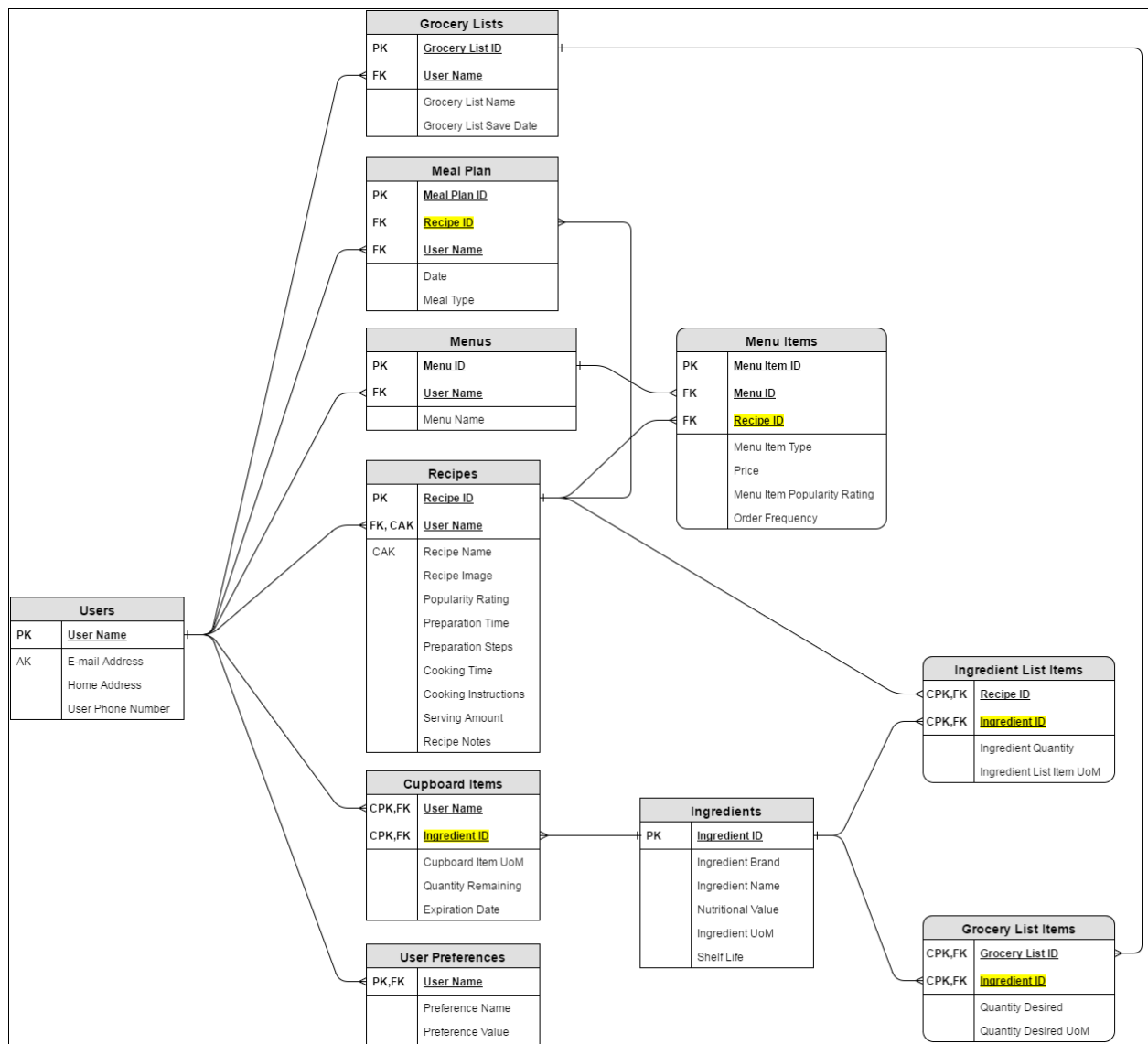


Figure 1: Entity Relationship Diagram with refined Foreign Keys

2.3 Relationship Characteristics

The next steps involve defining the degree and type of participation between the tables and defining the deletion rule between the tables. The degree of participation relates to the minimum and maximum number of references that one table can expect to have in a related table. The type of participation specifies which table is mandatory and which is optional within the scope of the table relationship. The deletion rule specifies the expected behavior of the child table when a delete request is made on the parent table. The results of this analysis for this database are illustrated in Figure 2.

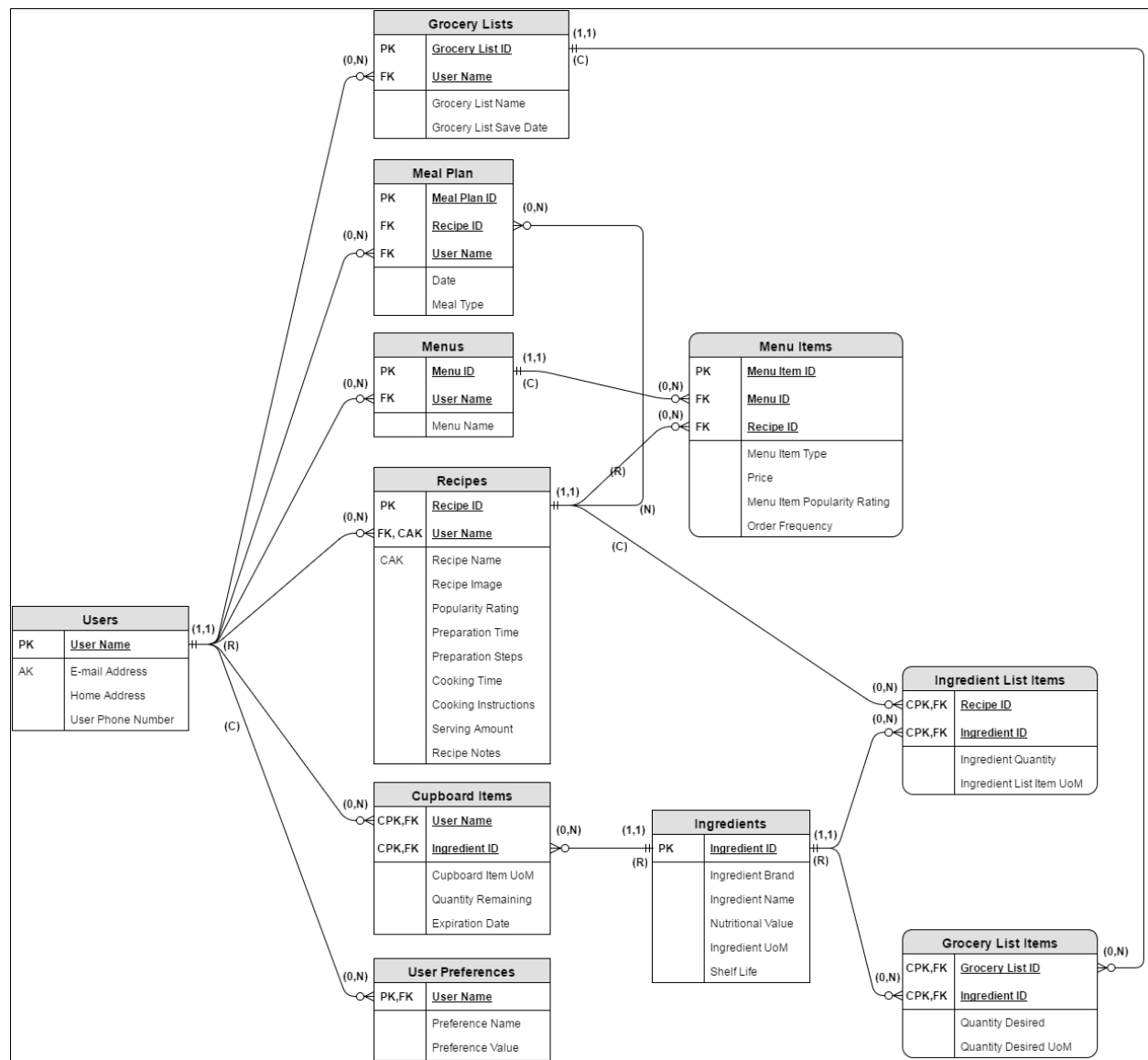


Figure 2: Entity Relationship Diagram with defined table relationships

2.4 Business Rules

Most of the business rules for the Recipe App database were specified in the field specifications of the previous project. While most business rules were determined to be application-oriented business rules, there was one business rule that was determined to be critical for the proper behavior of the database (and application). Each ingredient has a unit of measurement, but only a subset of all possible units of measure will be utilized by the application. To account for this, the units of measurement for each ingredient needs to be limited to supported types (ounces, grams, pounds, etc.). As seen in Figure 3, this is accomplished using a validation table named 'Measurement Units' with a single field named 'UoM ID'. This field serves as the primary key that each unit of measure field should be validated against. This also makes it very simple to add

support for more measurement types in the future. Each unit of measure field becomes a foreign key (highlighted in **yellow**) in order to establish the relationship to the validation table.

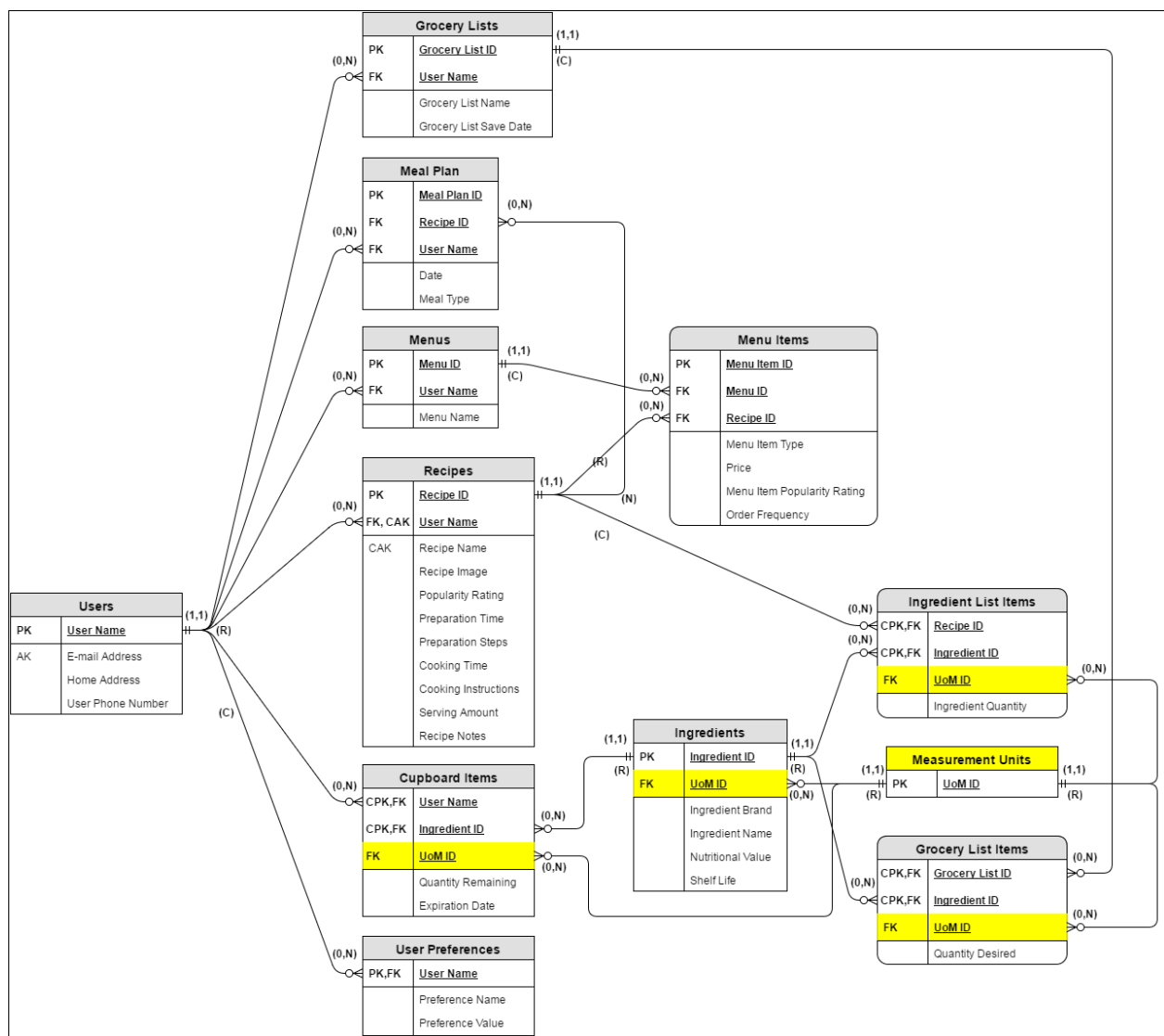


Figure 3: Entity Relationship diagram with Measurement Units validation table

With the relationships and characteristics defined for each table in the database, the design process is complete and the database is ready to be implemented and tested. The next section covers this topic in detail.

3. Database Implementation & Tests



This section provides the implementation details involved with realizing the database design (above). The SQL code used to create the tables is provided followed by tests for the tables. While these are not exhaustive tests, they do validate most of the critical details of the table structures.

3.1 Users

SQL Implementation:

```
CREATE TABLE `users` (
  `User_Name` VARCHAR(50) NOT NULL,
  `Email_Address` VARCHAR(100) NOT NULL,
  `Home_Address` VARCHAR(100) NULL DEFAULT NULL,
  `User_Phone_Number` VARCHAR(10) NULL DEFAULT NULL,
  PRIMARY KEY (`User_Name`),
  UNIQUE INDEX `E-mail Address` (`Email_Address`)
)
```

Tests:

Action	Expected	Actual								
Add a user with a user name only.	User cannot be created due to the E-mail Address being a required field.	<div>recipe_app.users: 0 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Email_Address</th><th>Home_Address</th><th>User_Phone_Number</th></tr><tr><td>user_01</td><td></td><td>NULL</td><td>NULL</td></tr></table> <div><div>Error</div><div> SQL Error (1364): Field 'Email_Address' doesn't have a default value</div><div>OK</div><div> Find some help on this error</div></div>	User_Name	Email_Address	Home_Address	User_Phone_Number	user_01		NULL	NULL
User_Name	Email_Address	Home_Address	User_Phone_Number							
user_01		NULL	NULL							

Action	Expected	Actual												
Add a user with an E-mail address only.	User cannot be created due to the User Name field being a required field.	<div>recipe_app.users: 0 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Email_Address</th><th>Home_Address</th><th>User_Phone_Number</th></tr><tr><td></td><td>user_01@aaa.com</td><td>(b)(3)</td><td>(b)(3)</td></tr></table> <div><div>Error</div><div><div><div></div></div><div>SQL Error (1364): Field 'User_Name' doesn't have a default value</div></div><div>OK</div><div><div>Find some help on this error</div></div></div>	User_Name	Email_Address	Home_Address	User_Phone_Number		user_01@aaa.com	(b)(3)	(b)(3)				
User_Name	Email_Address	Home_Address	User_Phone_Number											
	user_01@aaa.com	(b)(3)	(b)(3)											
Add a user with values for User Name and E-mail Address	User is successfully created and added to the database.	<div>recipe_app.users: 1 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Email_Address</th><th>Home_Address</th><th>User_Phone_Number</th></tr><tr><td>user_01</td><td>user_01@aaa.com</td><td>(b)(3)</td><td>(b)(3)</td></tr></table>	User_Name	Email_Address	Home_Address	User_Phone_Number	user_01	user_01@aaa.com	(b)(3)	(b)(3)				
User_Name	Email_Address	Home_Address	User_Phone_Number											
user_01	user_01@aaa.com	(b)(3)	(b)(3)											
Add another user with a User Name longer than 50 characters.	User cannot be created due to the 50 character constraint on the User Name field.	<div>recipe_app.users: 2 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Email_Address</th></tr><tr><td>user_01</td><td>user_01@aaa.com</td></tr><tr><td>123456789112345678921234567893123456789412345678951</td><td>xxx</td></tr></table> <div><div>Error</div><div><div><div></div></div><div>SQL Error (1406): Data too long for column 'User_Name' at row 1</div></div><div>OK</div><div><div>Find some help on this error</div></div></div>	User_Name	Email_Address	user_01	user_01@aaa.com	123456789112345678921234567893123456789412345678951	xxx						
User_Name	Email_Address													
user_01	user_01@aaa.com													
123456789112345678921234567893123456789412345678951	xxx													
Add the user again with valid values.	User is successfully created and added to the database.	<div>recipe_app.users: 2 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Email_Address</th><th>Home_Address</th><th>User_Phone_Number</th></tr><tr><td>user_01</td><td>user_01@aaa.com</td><td>(b)(3)</td><td>(b)(3)</td></tr><tr><td>user_02</td><td>user_02@zzz.com</td><td>123 American Way, City, St, Zip</td><td>1234567890</td></tr></table>	User_Name	Email_Address	Home_Address	User_Phone_Number	user_01	user_01@aaa.com	(b)(3)	(b)(3)	user_02	user_02@zzz.com	123 American Way, City, St, Zip	1234567890
User_Name	Email_Address	Home_Address	User_Phone_Number											
user_01	user_01@aaa.com	(b)(3)	(b)(3)											
user_02	user_02@zzz.com	123 American Way, City, St, Zip	1234567890											

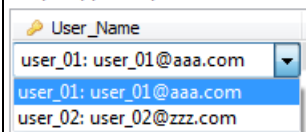
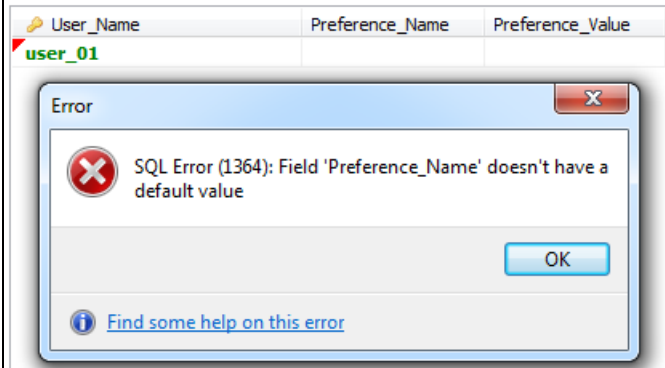
Action	Expected	Actual																
Add a third user using the same User Name value from the previously created user.	User cannot be created due to unique constraint on the User Name field.	<div>recipe_app.users: 2 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Email_Address</th><th>Home_Address</th><th>User_Phone_Number</th></tr><tr><td>user_01</td><td>user_01@aaa.com</td><td>(b)(6)</td><td>(b)(6)</td></tr><tr><td>user_02</td><td>user_02@zzz.com</td><td>123 American Way, City, St, Zip</td><td>1234567890</td></tr><tr><td>user_02</td><td>xxx</td><td>(b)(6)</td><td>(b)(6)</td></tr></table> <div><div>Error</div><div><div></div><div>SQL Error (1062): Duplicate entry 'user_02' for key 'PRIMARY'</div></div><div><div>OK</div></div><div><div></div><div>Find some help on this error</div></div></div>	User_Name	Email_Address	Home_Address	User_Phone_Number	user_01	user_01@aaa.com	(b)(6)	(b)(6)	user_02	user_02@zzz.com	123 American Way, City, St, Zip	1234567890	user_02	xxx	(b)(6)	(b)(6)
User_Name	Email_Address	Home_Address	User_Phone_Number															
user_01	user_01@aaa.com	(b)(6)	(b)(6)															
user_02	user_02@zzz.com	123 American Way, City, St, Zip	1234567890															
user_02	xxx	(b)(6)	(b)(6)															
Add a third user using a unique User Name value and the same E-mail Address value from the previously created user.	User cannot be created due to unique constraint on the E-mail Address field.	<div>recipe_app.users: 3 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Email_Address</th><th>Home_Address</th><th>User_Phone_Nu</th></tr><tr><td>user_01</td><td>user_01@aaa.com</td><td>(b)(6)</td><td>(b)(6)</td></tr><tr><td>user_02</td><td>user_02@zzz.com</td><td>123 American Way, City, St, Zip</td><td>1234567890</td></tr><tr><td>user_03</td><td>user_02@zzz.com</td><td>(b)(6)</td><td>(b)(6)</td></tr></table> <div><div>Error</div><div><div></div><div>SQL Error (1062): Duplicate entry 'user_02@zzz.com' for key 'E-mail Address'</div></div><div><div>OK</div></div><div><div></div><div>Find some help on this error</div></div></div>	User_Name	Email_Address	Home_Address	User_Phone_Nu	user_01	user_01@aaa.com	(b)(6)	(b)(6)	user_02	user_02@zzz.com	123 American Way, City, St, Zip	1234567890	user_03	user_02@zzz.com	(b)(6)	(b)(6)
User_Name	Email_Address	Home_Address	User_Phone_Nu															
user_01	user_01@aaa.com	(b)(6)	(b)(6)															
user_02	user_02@zzz.com	123 American Way, City, St, Zip	1234567890															
user_03	user_02@zzz.com	(b)(6)	(b)(6)															

3.2 User Preferences

SQL Implementation:

```
CREATE TABLE `user_preferences` (
  `User_Name` VARCHAR(50) NOT NULL,
  `Preference_Name` VARCHAR(50) NOT NULL,
  `Preference_Value` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`User_Name`),
  CONSTRAINT `FK_user_preferences_users` FOREIGN KEY (`User_Name`) REFERENCES `users` (`User_Name`) ON UPDATE
  CASCADE ON DELETE CASCADE
)
```

Tests:

Action	Expected	Actual
Add a record to User Preferences with only a User Name.	<p>Due to a foreign key constraint, the User Name field is limited to the values associated with the Users.User Name field.</p> <p>Record cannot be added due to Preference Name and Preference Value being required fields.</p>	<p>recipe_app.user_preferences: 0 rows total (approximately)</p>  <p>recipe_app.user_preferences: 0 rows total (approximately)</p> 

Action	Expected	Actual														
Add a record to User Preferences with only a User Name and Preference Name.	Record cannot be added due to Preference Value being a required field.	<div>recipe_app.user_preferences: 1 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Preference_Name</th><th>Preference_Value</th></tr><tr><td>user_01</td><td>Initials</td><td></td></tr></table> <div><div>Error</div><div><div></div><div>SQL Error (1364): Field 'Preference_Value' doesn't have a default value</div></div><div>OK</div><div><div></div><div>Find some help on this error</div></div></div>	User_Name	Preference_Name	Preference_Value	user_01	Initials									
User_Name	Preference_Name	Preference_Value														
user_01	Initials															
Add a record to User Preferences with all fields populated.	The record is added to User Preferences.	<div>recipe_app.user_preferences: 1 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Preference_Name</th><th>Preference_Value</th></tr><tr><td>user_01</td><td>Initials</td><td>u1</td></tr></table>	User_Name	Preference_Name	Preference_Value	user_01	Initials	u1								
User_Name	Preference_Name	Preference_Value														
user_01	Initials	u1														
Add a record to User Preferences with all fields populated for the second user.	The record is added to User Preferences.	<div>recipe_app.user_preferences: 2 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Preference_Name</th><th>Preference_Value</th></tr><tr><td>user_01</td><td>Initials</td><td>u1</td></tr><tr><td>user_02</td><td>Initials</td><td>u2</td></tr></table>	User_Name	Preference_Name	Preference_Value	user_01	Initials	u1	user_02	Initials	u2					
User_Name	Preference_Name	Preference_Value														
user_01	Initials	u1														
user_02	Initials	u2														
From the Users table, delete the second user from the database.	The second user in Users AND its referenced records in User Preferences are removed from the database due to the cascading deletion relationship between the two tables.	<div>recipe_app.users: 1 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Email_Address</th><th>Home_Address</th><th>User_Phone_Number</th></tr><tr><td>user_01</td><td>user_01@aaa.com</td><td>(NULL)</td><td>(NULL)</td></tr></table> <div>recipe_app.user_preferences: 1 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Preference_Name</th><th>Preference_Value</th></tr><tr><td>user_01</td><td>Initials</td><td>u1</td></tr></table>	User_Name	Email_Address	Home_Address	User_Phone_Number	user_01	user_01@aaa.com	(NULL)	(NULL)	User_Name	Preference_Name	Preference_Value	user_01	Initials	u1
User_Name	Email_Address	Home_Address	User_Phone_Number													
user_01	user_01@aaa.com	(NULL)	(NULL)													
User_Name	Preference_Name	Preference_Value														
user_01	Initials	u1														
Add a second user to the Users table.	A new user record is added to the Users table.	<div>recipe_app.users: 2 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Email_Address</th><th>Home_Address</th><th>User_Phone_Number</th></tr><tr><td>user_01</td><td>user_01@aaa.com</td><td>(NULL)</td><td>(NULL)</td></tr><tr><td>user_02</td><td>user_02@zzz.com</td><td>123 American Way, City, St, Zip</td><td>1234567890</td></tr></table>	User_Name	Email_Address	Home_Address	User_Phone_Number	user_01	user_01@aaa.com	(NULL)	(NULL)	user_02	user_02@zzz.com	123 American Way, City, St, Zip	1234567890		
User_Name	Email_Address	Home_Address	User_Phone_Number													
user_01	user_01@aaa.com	(NULL)	(NULL)													
user_02	user_02@zzz.com	123 American Way, City, St, Zip	1234567890													

3.3 Recipes & Ingredients

SQL Implementation:

```

CREATE TABLE `measurement_units` (
  `UoM_ID` VARCHAR(25) NOT NULL,
  PRIMARY KEY (`UoM_ID`)
)

CREATE TABLE `ingredients` (
  `Ingredient_ID` VARCHAR(25) NOT NULL,
  `Ingredient_Brand` VARCHAR(50) NULL DEFAULT NULL,
  `Ingredient_Name` VARCHAR(50) NOT NULL,
  `Nutritional_Value` MEDIUMINT(8) UNSIGNED NOT NULL DEFAULT '0',
  `UoM_ID` VARCHAR(25) NOT NULL DEFAULT 'unit',
  `Shelf_Life` MEDIUMINT(9) NULL DEFAULT NULL,
  PRIMARY KEY (`Ingredient_ID`),
  INDEX `FK_ingredients_measurement_units` (`UoM_ID`),
  CONSTRAINT `FK_ingredients_measurement_units` FOREIGN KEY (`UoM_ID`) REFERENCES `measurement_units`
  (`UoM_ID`) ON UPDATE CASCADE
)

CREATE TABLE `recipes` (
  `Recipe_ID` VARCHAR(25) NOT NULL,
  `User_Name` VARCHAR(50) NOT NULL,
  `Recipe_Name` VARCHAR(50) NOT NULL,
  `Recipe_Image` MEDIUMBLOB NULL,
  `Popularity_Rating` TINYINT(5) UNSIGNED NULL DEFAULT NULL,
  `Preparation_Time` SMALLINT(5) UNSIGNED NULL DEFAULT NULL,
  `Preparation_Steps` TEXT NULL,
  `Cooking_Time` SMALLINT(5) UNSIGNED NULL DEFAULT NULL,
  `Cooking_Instructions` TEXT NOT NULL,
  `Serving_Amount` FLOAT(5,2) UNSIGNED NOT NULL DEFAULT '0.00',
  `Recipe_Notes` TEXT NULL,
  PRIMARY KEY (`Recipe_ID`),
  UNIQUE INDEX `User Name_Recipe Name` (`User_Name`, `Recipe_Name`),
  CONSTRAINT `FK_recipes_users` FOREIGN KEY (`User_Name`) REFERENCES `users` (`User_Name`) ON UPDATE CASCADE
)

```

```

CREATE TABLE `ingredient_list_items` (
  `Recipe_ID` VARCHAR(25) NOT NULL,
  `Ingredient_ID` VARCHAR(25) NOT NULL,
  `Ingredient_Quantity` FLOAT(25,2) UNSIGNED NOT NULL DEFAULT '0.00',
  `UoM_ID` VARCHAR(25) NOT NULL DEFAULT 'unit',
  PRIMARY KEY (`Recipe_ID`, `Ingredient_ID`),
  INDEX `FK_ingredient_list_items_ingredients` (`Ingredient_ID`),
  INDEX `FK_ingredient_list_items_measurement_units` (`UoM_ID`),
  INDEX `Recipe_ID` (`Recipe_ID`),
  CONSTRAINT `FK_Ingredient_List_Items_recipes` FOREIGN KEY (`Recipe_ID`) REFERENCES `recipes` (`Recipe_ID`)
  ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT `FK_ingredient_list_items_ingredients` FOREIGN KEY (`Ingredient_ID`) REFERENCES `ingredients`
  (`Ingredient_ID`),
  CONSTRAINT `FK_ingredient_list_items_measurement_units` FOREIGN KEY (`UoM_ID`) REFERENCES
  `measurement_units` (`UoM_ID`) ON UPDATE CASCADE
)

```

Tests:

Action	Expected	Actual																						
Add a new record to the Recipes table providing values for Recipe ID, User Name, Recipe Name, and Cooking Instructions.	A new record is added to the Recipes table.	<p>recipe_app.recipes: 1 rows total (approximately)</p> <table><tr><th>Recipe_ID</th><th>User_Name</th><th>Recipe_Name</th><th>Recipe_Image</th></tr><tr><td>R1</td><td>user_01</td><td>R1 Name</td><td>0x89504E470...</td></tr></table> <table><tr><th>Popularity_Rating</th><th>Preparation_Time</th><th>Preparation_Steps</th><th>Cooking_Time</th></tr><tr><td>(NULL)</td><td>(NULL)</td><td>(NULL)</td><td>(NULL)</td></tr></table> <table><tr><th>Cooking_Instructions</th><th>Serving_Amount</th><th>Recipe_Notes</th></tr><tr><td>Step 1...</td><td>1.00</td><td>(NULL)</td></tr></table>	Recipe_ID	User_Name	Recipe_Name	Recipe_Image	R1	user_01	R1 Name	0x89504E470...	Popularity_Rating	Preparation_Time	Preparation_Steps	Cooking_Time	(NULL)	(NULL)	(NULL)	(NULL)	Cooking_Instructions	Serving_Amount	Recipe_Notes	Step 1...	1.00	(NULL)
Recipe_ID	User_Name	Recipe_Name	Recipe_Image																					
R1	user_01	R1 Name	0x89504E470...																					
Popularity_Rating	Preparation_Time	Preparation_Steps	Cooking_Time																					
(NULL)	(NULL)	(NULL)	(NULL)																					
Cooking_Instructions	Serving_Amount	Recipe_Notes																						
Step 1...	1.00	(NULL)																						
Create a duplicate of the record.	The record is not created due to the unique constraint created with the User Name and Recipe Name fields.	<p>recipe_app.recipes: 3 rows total (approximately)</p> <table><tr><th>Recipe_ID</th><th>User_Name</th><th>Recipe_Name</th><th>Recipe_Image</th></tr><tr><td>R1</td><td>user_01</td><td>R1 Name</td><td>0x89504E47...</td></tr><tr><td>R4</td><td>user_01</td><td>R1 Name</td><td>0x89504E47...</td></tr></table> <div><div>Error</div><div><div></div><div>SQL Error (1062): Duplicate entry 'user_01-R1 Name' for key 'User Name_Recipe Name'</div></div></div>	Recipe_ID	User_Name	Recipe_Name	Recipe_Image	R1	user_01	R1 Name	0x89504E47...	R4	user_01	R1 Name	0x89504E47...										
Recipe_ID	User_Name	Recipe_Name	Recipe_Image																					
R1	user_01	R1 Name	0x89504E47...																					
R4	user_01	R1 Name	0x89504E47...																					

Action	Expected	Actual																								
From the Users table, delete the user associated with the previously created Recipe record.	The user is not deleted due to the restrictive deletion relationship between the tables.	<div>recipe_app.users: 2 rows total (approximately)</div> <table><tr><th>User_Name</th><th>Email_Address</th><th>Home_Address</th><th>Use</th></tr><tr><td>user_01</td><td>user_01@aaa.com</td><td>(NULL)</td><td>(N</td></tr><tr><td>user_02</td><td></td><td></td><td>12</td></tr></table> <div>Error</div> <div><div></div><div>Grid editing error</div><div>SQL Error (1451): Cannot delete or update a parent row: a foreign key constraint fails ('recipe_app'.recipes', CONSTRAINT 'FK_recipes_users' FOREIGN KEY ('User_Name') REFERENCES 'users' ('User_Name') ON UPDATE CASCADE)</div><div>OK</div></div>	User_Name	Email_Address	Home_Address	Use	user_01	user_01@aaa.com	(NULL)	(N	user_02			12												
User_Name	Email_Address	Home_Address	Use																							
user_01	user_01@aaa.com	(NULL)	(N																							
user_02			12																							
Show the Measurement Units validation table.	The table contains a single field named UoM ID with the following values: cup, gram, liter, ounce, pound, quart, unit.	<div>recipe_app.measurement_units: 7 rows total</div> <table><tr><th>UoM_ID</th></tr><tr><td>cup</td></tr><tr><td>gram</td></tr><tr><td>liter</td></tr><tr><td>ounce</td></tr><tr><td>pound</td></tr><tr><td>quart</td></tr><tr><td>unit</td></tr></table>	UoM_ID	cup	gram	liter	ounce	pound	quart	unit																
UoM_ID																										
cup																										
gram																										
liter																										
ounce																										
pound																										
quart																										
unit																										
Add a record to the Ingredients table for water specifying ounces for UoM.	<div>The UoM ID field only allows values from the Measurement Units validation table.</div> <div>The record is added to the Ingredients table.</div>	<div>recipe_app.ingredients: 0 rows total (approximately)</div> <table><tr><th>Ingredient_ID</th><th>Ingredient_Brand</th><th>Ingredient_Name</th><th>Nutritional_Value</th><th>UoM_ID</th><th>Shelf_Life</th></tr><tr><td>11</td><td>(NULL)</td><td>Water</td><td>0</td><td><div>unit</div><div>cup</div><div>gram</div><div>liter</div><div>ounce</div><div>pound</div><div>quart</div><div>unit</div></td><td>(NULL)</td></tr></table> <div>recipe_app.ingredients: 1 rows total (approximately)</div> <table><tr><th>Ingredient_ID</th><th>Ingredient_Brand</th><th>Ingredient_Name</th><th>Nutritional_Value</th><th>UoM_ID</th><th>Shelf_Life</th></tr><tr><td>11</td><td>(NULL)</td><td>Water</td><td>0</td><td>ounce</td><td>(NULL)</td></tr></table>	Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	UoM_ID	Shelf_Life	11	(NULL)	Water	0	<div>unit</div> <div>cup</div> <div>gram</div> <div>liter</div> <div>ounce</div> <div>pound</div> <div>quart</div> <div>unit</div>	(NULL)	Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	UoM_ID	Shelf_Life	11	(NULL)	Water	0	ounce	(NULL)
Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	UoM_ID	Shelf_Life																					
11	(NULL)	Water	0	<div>unit</div> <div>cup</div> <div>gram</div> <div>liter</div> <div>ounce</div> <div>pound</div> <div>quart</div> <div>unit</div>	(NULL)																					
Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	UoM_ID	Shelf_Life																					
11	(NULL)	Water	0	ounce	(NULL)																					

Action	Expected	Actual																								
Add two records for bread and mustard, respectively.	The records are successfully added to the Ingredients table.	<div>recipe_app.ingredients: 3 rows total (approximately)</div> <table><tr><th>Ingredient_ID</th><th>Ingredient_Brand</th><th>Ingredient_Name</th><th>Nutritional_Value</th><th>UoM_ID</th><th>Shelf_Life</th></tr><tr><td>I1</td><td>(NULL)</td><td>Water</td><td>0</td><td>ounce</td><td>(NULL)</td></tr><tr><td>I2</td><td>(NULL)</td><td>Bread</td><td>100</td><td>unit</td><td>14</td></tr><tr><td>I3</td><td>(NULL)</td><td>Mustard</td><td>30</td><td>gram</td><td>(NULL)</td></tr></table>	Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	UoM_ID	Shelf_Life	I1	(NULL)	Water	0	ounce	(NULL)	I2	(NULL)	Bread	100	unit	14	I3	(NULL)	Mustard	30	gram	(NULL)
Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	UoM_ID	Shelf_Life																					
I1	(NULL)	Water	0	ounce	(NULL)																					
I2	(NULL)	Bread	100	unit	14																					
I3	(NULL)	Mustard	30	gram	(NULL)																					
From the Ingredient List Items table, create new records associating them with the created recipe and the water and bread ingredients, respectively.	The records are added to the Ingredients List Item table.	<div>recipe_app.ingredient_list_items: 0 rows total (approximately)</div> <div><table><tr><th>Recipe_ID</th><th>Ingredient_ID</th><th>Ingredient_Quantity</th><th>UoM_ID</th></tr><tr><td></td><td>I1:</td><td>0.00</td><td>unit</td></tr></table><div><div>R1: user_01</div><div>I1:</div><div>I2:</div><div>I3:</div></div></div> <div>recipe_app.ingredient_list_items: 2 rows total (approximately)</div> <table><tr><th>Recipe_ID</th><th>Ingredient_ID</th><th>Ingredient_Quantity</th><th>UoM_ID</th></tr><tr><td>R1</td><td>I1</td><td>5.00</td><td>ounce</td></tr><tr><td>R1</td><td>I2</td><td>1.00</td><td>unit</td></tr></table>	Recipe_ID	Ingredient_ID	Ingredient_Quantity	UoM_ID		I1:	0.00	unit	Recipe_ID	Ingredient_ID	Ingredient_Quantity	UoM_ID	R1	I1	5.00	ounce	R1	I2	1.00	unit				
Recipe_ID	Ingredient_ID	Ingredient_Quantity	UoM_ID																							
	I1:	0.00	unit																							
Recipe_ID	Ingredient_ID	Ingredient_Quantity	UoM_ID																							
R1	I1	5.00	ounce																							
R1	I2	1.00	unit																							
Execute the following SQL query: <pre>select r.Recipe_Name, r.Cooking_Instructions, li.Ingredient_Quantity, li.UoM_ID, i.Ingredient_Name from recipes r, ingredient_list_items li, ingredients i where li.Recipe_ID = r.Recipe_ID and i.Ingredient_ID = li.Ingredient_ID</pre>	A user-friendly view of the recipe and its ingredients are shown in a view table.	<pre>1 select r.Recipe_Name, r.Cooking_Instructions, 2 li.Ingredient_Quantity, li.UoM_ID, i.Ingredient_Name 3 from recipes r, ingredient_list_items li, ingredients i 4 where li.Recipe_ID = r.Recipe_ID 5 and i.Ingredient_ID = li.Ingredient_ID</pre> <div>Result #1 (5x2)</div> <table><tr><th>Recipe_Name</th><th>Cooking_Instructions</th><th>Ingredient_Quantity</th><th>UoM_ID</th><th>Ingredient_Name</th></tr><tr><td>R1Name</td><td>Step 1...</td><td>5.00</td><td>ounce</td><td>Water</td></tr><tr><td>R1Name</td><td>Step 1...</td><td>1.00</td><td>unit</td><td>Bread</td></tr></table>	Recipe_Name	Cooking_Instructions	Ingredient_Quantity	UoM_ID	Ingredient_Name	R1Name	Step 1...	5.00	ounce	Water	R1Name	Step 1...	1.00	unit	Bread									
Recipe_Name	Cooking_Instructions	Ingredient_Quantity	UoM_ID	Ingredient_Name																						
R1Name	Step 1...	5.00	ounce	Water																						
R1Name	Step 1...	1.00	unit	Bread																						

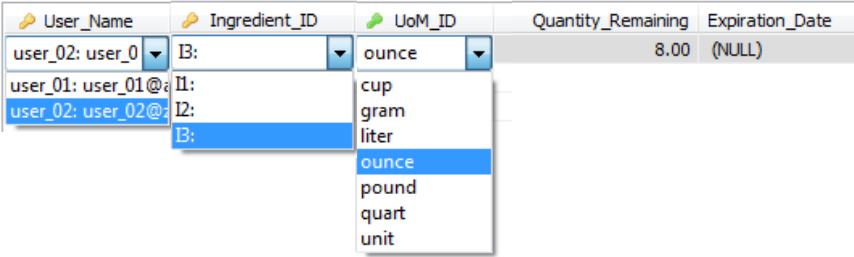
Action	Expected	Actual																		
From the Ingredients table, delete the water ingredient.	The record cannot be deleted due to the restrict deletion relationship between the Ingredients and Ingredient List Items tables.	<div>recipe_app.ingredients: 3 rows total (approximately)</div> <table><tr><th>Ingredient_ID</th><th>Ingredient_Brand</th><th>Ingredient_Name</th><th>Nutritional_Value</th></tr><tr><td>11</td><td>(NULL)</td><td>Water</td><td></td></tr></table> <div>Error<div><div></div><div>Grid editing error</div><div>SQL Error (1451): Cannot delete or update a parent row: a foreign key constraint fails ('recipe_app`.`ingredient_list_items`, CONSTRAINT `FK_ingredient_list_items_ingredients` FOREIGN KEY (`Ingredient_ID`) REFERENCES `ingredients` (`Ingredient_ID`))</div></div></div>	Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	11	(NULL)	Water											
Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value																	
11	(NULL)	Water																		
From the Recipes table, delete the created recipe record.	The recipe in Recipes AND its referenced records in Ingredient List Items are removed from the database due to the cascading deletion relationship between the two tables. After deletion, both tables are empty.	<div>recipe_app.recipes: 0 rows total (approximately)</div> <table><tr><th>Recipe_ID</th><th>User_Name</th><th>Recipe_Name</th><th>Recipe_Image</th></tr></table> <div>recipe_app.ingredient_list_items: 0 rows total (approximately)</div> <table><tr><th>Recipe_ID</th><th>Ingredient_ID</th><th>Ingredient_Quantity</th><th>UoM_ID</th></tr></table>	Recipe_ID	User_Name	Recipe_Name	Recipe_Image	Recipe_ID	Ingredient_ID	Ingredient_Quantity	UoM_ID										
Recipe_ID	User_Name	Recipe_Name	Recipe_Image																	
Recipe_ID	Ingredient_ID	Ingredient_Quantity	UoM_ID																	
From the Ingredients table, delete the water ingredient.	The ingredient can now be deleted since there are no references to this record.	<div>recipe_app.ingredients: 2 rows total (approximately)</div> <table><tr><th>Ingredient_ID</th><th>Ingredient_Brand</th><th>Ingredient_Name</th><th>Nutritional_Value</th><th>UoM_ID</th><th>Shelf_Life</th></tr><tr><td>12</td><td>(NULL)</td><td>Bread</td><td>100</td><td>unit</td><td>14</td></tr><tr><td>13</td><td>(NULL)</td><td>Mustard</td><td>30</td><td>gram</td><td>(NULL)</td></tr></table>	Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	UoM_ID	Shelf_Life	12	(NULL)	Bread	100	unit	14	13	(NULL)	Mustard	30	gram	(NULL)
Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	UoM_ID	Shelf_Life															
12	(NULL)	Bread	100	unit	14															
13	(NULL)	Mustard	30	gram	(NULL)															

3.4 Cupboard Items

SQL Implementation:

```
CREATE TABLE `cupboard_items` (
  `User_Name` VARCHAR(50) NOT NULL,
  `Ingredient_ID` VARCHAR(25) NOT NULL,
  `UoM_ID` VARCHAR(25) NOT NULL DEFAULT 'unit',
  `Quantity_Remaining` DOUBLE(24,2) UNSIGNED NULL DEFAULT '0.00',
  `Expiration_Date` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`User_Name`, `Ingredient_ID`),
  INDEX `FK_cupboard_items_ingredients` (`Ingredient_ID`),
  INDEX `FK_cupboard_items_measurement_units` (`UoM_ID`),
  CONSTRAINT `FK_cupboard_items_ingredients` FOREIGN KEY (`Ingredient_ID`) REFERENCES `ingredients`
  (`Ingredient_ID`),
  CONSTRAINT `FK_cupboard_items_users` FOREIGN KEY (`User_Name`) REFERENCES `users` (`User_Name`) ON UPDATE
  CASCADE,
  CONSTRAINT `FK_cupboard_items_measurement_units` FOREIGN KEY (`UoM_ID`) REFERENCES `measurement_units`
  (`UoM_ID`)
)
```

Tests:

Action	Expected	Actual
For the second user, add the mustard ingredient (I3) to the Cupboard Items table.	The User Name field is limited to the existing records in the Users table; the Ingredient ID field is limited to the existing records in the Ingredients table; the UoM ID field is limited to the existing values in the Measurement Units validation table; the record is successfully added to the Cupboard Items table.	<p>recipe_app.cupboard_items: 3 rows total (approximately)</p> 

Action	Expected	Actual																				
For the first user, add the water (I1) and mustard (I3) ingredient to the Cupboard Items table.	The records are successfully added to the Cupboard Items table.	<div>recipe_app.cupboard_items: 3 rows total (approximately)</div> <table><thead><tr><th>User_Name</th><th>Ingredient_ID</th><th>UoM_ID</th><th>Quantity_Remaining</th><th>Expiration_Date</th></tr></thead><tbody><tr><td>user_02</td><td>I3</td><td>ounce</td><td>8.00</td><td>(NULL)</td></tr><tr><td>user_01</td><td>I1</td><td>ounce</td><td>64.00</td><td>(NULL)</td></tr><tr><td>user_01</td><td>I3</td><td>gram</td><td>285.00</td><td>2016-06-24</td></tr></tbody></table>	User_Name	Ingredient_ID	UoM_ID	Quantity_Remaining	Expiration_Date	user_02	I3	ounce	8.00	(NULL)	user_01	I1	ounce	64.00	(NULL)	user_01	I3	gram	285.00	2016-06-24
User_Name	Ingredient_ID	UoM_ID	Quantity_Remaining	Expiration_Date																		
user_02	I3	ounce	8.00	(NULL)																		
user_01	I1	ounce	64.00	(NULL)																		
user_01	I3	gram	285.00	2016-06-24																		
For the second user, add the mustard ingredient (I3) with different units to the Cupboard Items table.	The record cannot be added due to the composite primary key constraint created with the User Name and Ingredient ID fields.	<div>recipe_app.cupboard_items: 3 rows total (approximately)</div> <table><thead><tr><th>User_Name</th><th>Ingredient_ID</th><th>UoM_ID</th><th>Quantity_Remaining</th></tr></thead><tbody><tr><td>user_02</td><td>I3</td><td>ounce</td><td>8.00</td></tr><tr><td>user_02</td><td>I3</td><td>gram</td><td>253</td></tr><tr><td></td><td></td><td></td><td>64.00</td></tr><tr><td></td><td></td><td></td><td>285.00</td></tr></tbody></table> <div>Error<div>SQL Error (1062): Duplicate entry 'user_02-I3' for key 'PRIMARY'</div></div>	User_Name	Ingredient_ID	UoM_ID	Quantity_Remaining	user_02	I3	ounce	8.00	user_02	I3	gram	253				64.00				285.00
User_Name	Ingredient_ID	UoM_ID	Quantity_Remaining																			
user_02	I3	ounce	8.00																			
user_02	I3	gram	253																			
			64.00																			
			285.00																			
From the Ingredients table, delete the water ingredient.	The record cannot be deleted due to the restrict deletion relationship between the Ingredients and Cupboard Items tables.	<div>recipe_app.ingredients: 3 rows total (approximately)</div> <table><thead><tr><th>Ingredient_ID</th><th>Ingredient_Brand</th><th>Ingredient_Name</th><th>Nutritional_Value</th><th>UoM_ID</th></tr></thead><tbody><tr><td>I1</td><td>(NULL)</td><td>Water</td><td>0</td><td>ounce</td></tr><tr><td>I2</td><td></td><td></td><td>100</td><td>unit</td></tr><tr><td>I3</td><td></td><td></td><td>30</td><td>gram</td></tr></tbody></table> <div>Error<div>Grid editing error<div>SQL Error (1451): Cannot delete or update a parent row: a foreign key constraint fails ('recipe_app`.`cupboard_items`, CONSTRAINT 'FK_cupboard_items_ingredients' FOREIGN KEY ('Ingredient_ID') REFERENCES 'ingredients' ('Ingredient_ID'))</div></div></div>	Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	UoM_ID	I1	(NULL)	Water	0	ounce	I2			100	unit	I3			30	gram
Ingredient_ID	Ingredient_Brand	Ingredient_Name	Nutritional_Value	UoM_ID																		
I1	(NULL)	Water	0	ounce																		
I2			100	unit																		
I3			30	gram																		

3.5 Menus

SQL Implementation:

```
CREATE TABLE `menus` (  
  `Menu_ID` VARCHAR(25) NOT NULL,  
  `User_Name` VARCHAR(50) NOT NULL,  
  `Menu_Name` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`Menu_ID`),  
  INDEX `FK_menus_users` (`User_Name`),  
  CONSTRAINT `FK_menus_users` FOREIGN KEY (`User_Name`) REFERENCES `users` (`User_Name`) ON UPDATE CASCADE  
)  
  
CREATE TABLE `menu_items` (  
  `Menu_Item_ID` VARCHAR(25) NOT NULL,  
  `Menu_ID` VARCHAR(25) NOT NULL,  
  `Recipe_ID` VARCHAR(25) NOT NULL,  
  `Menu_Item_Type` VARCHAR(25) NOT NULL,  
  `Price` FLOAT(9,2) UNSIGNED NULL DEFAULT NULL,  
  `Menu_Item_Popularity_Rating` TINYINT(5) UNSIGNED NULL DEFAULT NULL,  
  `Order_Frequency` FLOAT(5,2) UNSIGNED NULL DEFAULT NULL,  
  PRIMARY KEY (`Menu_Item_ID`),  
  INDEX `FK_menu_items_recipes` (`Recipe_ID`),  
  INDEX `FK_menu_items_menus` (`Menu_ID`),  
  CONSTRAINT `FK_menu_items_recipes` FOREIGN KEY (`Recipe_ID`) REFERENCES `recipes` (`Recipe_ID`),  
  CONSTRAINT `FK_menu_items_menus` FOREIGN KEY (`Menu_ID`) REFERENCES `menus` (`Menu_ID`) ON UPDATE CASCADE ON  
  DELETE CASCADE  
)
```


Tests:

Action	Expected	Actual																																												
For the second user, add a new recipe to the Recipes table, and then create a duplicate of the new recipe for the first user.	Two recipe records are successfully added to the Recipes table.	<div>recipe_app.recipes: 3 rows total (approximately)</div> <table><tr><th>Recipe_ID</th><th>User_Name</th><th>Recipe_Name</th><th>Recipe_Image</th><th>Popularity_Rating</th><th>Preparation_Time</th></tr><tr><td>R1</td><td>user_01</td><td>R1 Name</td><td>0x89504E47...</td><td>(NULL)</td><td>(NULL)</td></tr><tr><td>R2</td><td>user_02</td><td>R2 Name</td><td>(NULL)</td><td>(NULL)</td><td>10</td></tr><tr><td>R3</td><td>user_01</td><td>R2 Name</td><td>(NULL)</td><td>(NULL)</td><td>10</td></tr></table> <div><table><tr><th>Preparation_Steps</th><th>Cooking_Time</th><th>Cooking_Instructions</th><th>Serving_Amount</th><th>Recipe_Notes</th></tr><tr><td>(NULL)</td><td>(NULL)</td><td>Step 1...</td><td>1.00</td><td>(NULL)</td></tr><tr><td>(NULL)</td><td>30</td><td>Step 1...</td><td>1.00</td><td>Add salt!!</td></tr><tr><td>(NULL)</td><td>30</td><td>Step 1...</td><td>1.00</td><td>Add salt!!</td></tr></table></div>	Recipe_ID	User_Name	Recipe_Name	Recipe_Image	Popularity_Rating	Preparation_Time	R1	user_01	R1 Name	0x89504E47...	(NULL)	(NULL)	R2	user_02	R2 Name	(NULL)	(NULL)	10	R3	user_01	R2 Name	(NULL)	(NULL)	10	Preparation_Steps	Cooking_Time	Cooking_Instructions	Serving_Amount	Recipe_Notes	(NULL)	(NULL)	Step 1...	1.00	(NULL)	(NULL)	30	Step 1...	1.00	Add salt!!	(NULL)	30	Step 1...	1.00	Add salt!!
Recipe_ID	User_Name	Recipe_Name	Recipe_Image	Popularity_Rating	Preparation_Time																																									
R1	user_01	R1 Name	0x89504E47...	(NULL)	(NULL)																																									
R2	user_02	R2 Name	(NULL)	(NULL)	10																																									
R3	user_01	R2 Name	(NULL)	(NULL)	10																																									
Preparation_Steps	Cooking_Time	Cooking_Instructions	Serving_Amount	Recipe_Notes																																										
(NULL)	(NULL)	Step 1...	1.00	(NULL)																																										
(NULL)	30	Step 1...	1.00	Add salt!!																																										
(NULL)	30	Step 1...	1.00	Add salt!!																																										
For the first user, add a menu named “Menu U1” to the Menus table.	The User Name field is limited to the existing records in the Users table; the record is successfully added to the Menus table.	<div>recipe_app.menus: 1 rows total (approximately)</div> <table><tr><th>Menu_ID</th><th>User_Name</th><th>Menu_Name</th></tr><tr><td>M1</td><td>user_01: user_01@</td><td>Menu U1</td></tr></table>	Menu_ID	User_Name	Menu_Name	M1	user_01: user_01@	Menu U1																																						
Menu_ID	User_Name	Menu_Name																																												
M1	user_01: user_01@	Menu U1																																												
From the Menu Items table, add a record associating it with the created menu and the R1 recipe.	The Menu ID field is limited to the existing records in the Menus table; the Recipe ID field is limited to the existing records in the Recipes table; the menu item is successfully added to the table.	<div>recipe_app.menu_items: 1 rows total (approximately)</div> <table><tr><th>Menu_Item_ID</th><th>Menu_ID</th><th>Recipe_ID</th><th>Menu_Item_Type</th><th>Price</th><th>Menu_Item_Popularity_Rating</th><th>Order_Frequency</th></tr><tr><td>MI1</td><td>M1: user_01</td><td>R1: user_01</td><td>Breakfast</td><td>(NULL)</td><td>(NULL)</td><td>(NULL)</td></tr></table>	Menu_Item_ID	Menu_ID	Recipe_ID	Menu_Item_Type	Price	Menu_Item_Popularity_Rating	Order_Frequency	MI1	M1: user_01	R1: user_01	Breakfast	(NULL)	(NULL)	(NULL)																														
Menu_Item_ID	Menu_ID	Recipe_ID	Menu_Item_Type	Price	Menu_Item_Popularity_Rating	Order_Frequency																																								
MI1	M1: user_01	R1: user_01	Breakfast	(NULL)	(NULL)	(NULL)																																								
DESIGN NOTE: The degree of participation should be modified to enforce one Menu Item must exist per Menu. Also, restrictions should be in place such that only recipes available to the owner of the menu are available as menu items.																																														

Action	Expected	Actual																					
From the Recipes table, delete the R1 recipe record.	The record cannot be deleted due to the restrict deletion relationship between the Recipes and Menu Items tables.	<div>recipe_app.recipes: 3 rows total (approximately)</div> <table><tr><th>Recipe_ID</th><th>User_Name</th><th>Recipe_Name</th><th>Recipe_Image</th></tr><tr><td>R1</td><td>user_01</td><td>R1 Name</td><td>0x89504E47</td></tr></table> <div>Error</div> <div><div></div><div>Grid editing error</div><div>SQL Error (1451): Cannot delete or update a parent row: a foreign key constraint fails ('recipe_app`.`menu_items`, CONSTRAINT 'FK_menu_items_recipes' FOREIGN KEY ('Recipe_ID') REFERENCES 'recipes' ('Recipe_ID'))</div></div>	Recipe_ID	User_Name	Recipe_Name	Recipe_Image	R1	user_01	R1 Name	0x89504E47													
Recipe_ID	User_Name	Recipe_Name	Recipe_Image																				
R1	user_01	R1 Name	0x89504E47																				
For the second user, add a new menu to the Menus table and create new menu items for it in the Menu Items table.	The new menu record is successfully created; the new menu item is successfully created and associated with the menu.	<div>recipe_app.menus: 2 rows total (approximately)</div> <table><tr><th>Menu_ID</th><th>User_Name</th><th>Menu_Name</th></tr><tr><td>M1</td><td>user_01</td><td>Menu U1</td></tr><tr><td>M2</td><td>user_02</td><td>Menu U2</td></tr></table> <div>recipe_app.menu_items: 2 rows total (approximately)</div> <table><tr><th>Menu_Item_ID</th><th>Menu_ID</th><th>Recipe_ID</th><th>Menu_Item_Type</th></tr><tr><td>MI1</td><td>M1</td><td>R1</td><td>Breakfast</td></tr><tr><td>MI2</td><td>M2</td><td>R2</td><td>Brunch</td></tr></table>	Menu_ID	User_Name	Menu_Name	M1	user_01	Menu U1	M2	user_02	Menu U2	Menu_Item_ID	Menu_ID	Recipe_ID	Menu_Item_Type	MI1	M1	R1	Breakfast	MI2	M2	R2	Brunch
Menu_ID	User_Name	Menu_Name																					
M1	user_01	Menu U1																					
M2	user_02	Menu U2																					
Menu_Item_ID	Menu_ID	Recipe_ID	Menu_Item_Type																				
MI1	M1	R1	Breakfast																				
MI2	M2	R2	Brunch																				
From the Menus table, delete the M1 menu.	The M1 record is deleted from the Menus table; due to the cascading deletion relationship between Menus and Menu Items, the MI1 record is deleted from the Menu Items table; only the menu and menu item records remain for the second user.	<div>recipe_app.menus: 1 rows total (approximately)</div> <table><tr><th>Menu_ID</th><th>User_Name</th><th>Menu_Name</th></tr><tr><td>M2</td><td>user_02</td><td>Menu U2</td></tr></table> <div>recipe_app.menu_items: 1 rows total (approximately)</div> <table><tr><th>Menu_Item_ID</th><th>Menu_ID</th><th>Recipe_ID</th><th>Menu_Item_Type</th></tr><tr><td>MI2</td><td>M2</td><td>R2</td><td>Brunch</td></tr></table>	Menu_ID	User_Name	Menu_Name	M2	user_02	Menu U2	Menu_Item_ID	Menu_ID	Recipe_ID	Menu_Item_Type	MI2	M2	R2	Brunch							
Menu_ID	User_Name	Menu_Name																					
M2	user_02	Menu U2																					
Menu_Item_ID	Menu_ID	Recipe_ID	Menu_Item_Type																				
MI2	M2	R2	Brunch																				

3.6 Meal Plan

SQL Implementation:

```
CREATE TABLE `meal_plan` (
  `Meal_Plan_ID` VARCHAR(25) NOT NULL,
  `User_Name` VARCHAR(50) NOT NULL,
  `Recipe_ID` VARCHAR(25) NULL DEFAULT NULL,
  `Meal_Type` VARCHAR(25) NOT NULL,
  `Date` DATE NOT NULL,
  PRIMARY KEY (`Meal_Plan_ID`),
  INDEX `FK_recipes` (`Recipe_ID`),
  INDEX `FK_meal_plan_users` (`User_Name`),
  CONSTRAINT `FK_recipes` FOREIGN KEY (`Recipe_ID`) REFERENCES `recipes` (`Recipe_ID`) ON UPDATE CASCADE ON DELETE SET NULL,
  CONSTRAINT `FK_meal_plan_users` FOREIGN KEY (`User_Name`) REFERENCES `users` (`User_Name`) ON UPDATE CASCADE
)
```

Tests:

Action	Expected	Actual																														
For the first user, add a new meal plan for June 24, 2015.	The User Name field is limited to the existing records in the Users table; the Recipe ID is limited to the existing records in the Recipes table; the record is successfully added.	<div>recipe_app.meal_plan: 1 rows total (approximately)</div> <table><tr><th>Meal_Plan_ID</th><th>User_Name</th><th>Recipe_ID</th><th>Meal_Type</th><th>Date</th></tr><tr><td>MP1</td><td>user_01: user_01</td><td>R1: user_01</td><td>Breakfast</td><td>2015-06-24</td></tr></table>	Meal_Plan_ID	User_Name	Recipe_ID	Meal_Type	Date	MP1	user_01: user_01	R1: user_01	Breakfast	2015-06-24																				
Meal_Plan_ID	User_Name	Recipe_ID	Meal_Type	Date																												
MP1	user_01: user_01	R1: user_01	Breakfast	2015-06-24																												
Add more meal plan records for both users using different dates between the users.	The records are successfully added to the Meal Plan table.	<div>recipe_app.meal_plan: 5 rows total (approximately)</div> <table><tr><th>Meal_Plan_ID</th><th>User_Name</th><th>Recipe_ID</th><th>Meal_Type</th><th>Date</th></tr><tr><td>MP1</td><td>user_01</td><td>R1</td><td>Breakfast</td><td>2015-06-24</td></tr><tr><td>MP2</td><td>user_01</td><td>R3</td><td>Dinner</td><td>2015-06-24</td></tr><tr><td>MP3</td><td>user_02</td><td>R2</td><td>Snack</td><td>2016-03-01</td></tr><tr><td>MP4</td><td>user_01</td><td>R1</td><td>Snack</td><td>2015-06-24</td></tr><tr><td>MP5</td><td>user_02</td><td>R2</td><td>Lunch</td><td>2016-03-01</td></tr></table>	Meal_Plan_ID	User_Name	Recipe_ID	Meal_Type	Date	MP1	user_01	R1	Breakfast	2015-06-24	MP2	user_01	R3	Dinner	2015-06-24	MP3	user_02	R2	Snack	2016-03-01	MP4	user_01	R1	Snack	2015-06-24	MP5	user_02	R2	Lunch	2016-03-01
Meal_Plan_ID	User_Name	Recipe_ID	Meal_Type	Date																												
MP1	user_01	R1	Breakfast	2015-06-24																												
MP2	user_01	R3	Dinner	2015-06-24																												
MP3	user_02	R2	Snack	2016-03-01																												
MP4	user_01	R1	Snack	2015-06-24																												
MP5	user_02	R2	Lunch	2016-03-01																												

Action	Expected	Actual																																										
From the Recipes table, delete the R3 recipe record.	The R3 record is deleted from the Recipes table; the Meal Plan.Recipe ID field for record MP2 is changed to NULL due to the null deletion relationship between the Recipes and Meal Plan tables.	<div>recipe_app.recipes: 2 rows total (approximately)</div> <table><tr><th>Recipe_ID</th><th>User_Name</th><th>Recipe_Name</th><th>Recipe_Ima</th></tr><tr><td>R1</td><td>user_01</td><td>R1 Name</td><td>0x89504E4</td></tr><tr><td>R2</td><td>user_02</td><td>R2 Name</td><td>(NULL)</td></tr></table> <div>recipe_app.meal_plan: 5 rows total (approximately)</div> <table><tr><th>Meal_Plan_ID</th><th>User_Name</th><th>Recipe_ID</th><th>Meal_Type</th><th>Date</th></tr><tr><td>MP1</td><td>user_01</td><td>R1</td><td>Breakfast</td><td>2015-06-24</td></tr><tr><td>MP2</td><td>user_01</td><td>(NULL)</td><td>Dinner</td><td>2015-06-24</td></tr><tr><td>MP3</td><td>user_02</td><td>R2</td><td>Snack</td><td>2016-03-01</td></tr><tr><td>MP4</td><td>user_01</td><td>R1</td><td>Snack</td><td>2015-06-24</td></tr><tr><td>MP5</td><td>user_02</td><td>R2</td><td>Lunch</td><td>2016-03-01</td></tr></table>	Recipe_ID	User_Name	Recipe_Name	Recipe_Ima	R1	user_01	R1 Name	0x89504E4	R2	user_02	R2 Name	(NULL)	Meal_Plan_ID	User_Name	Recipe_ID	Meal_Type	Date	MP1	user_01	R1	Breakfast	2015-06-24	MP2	user_01	(NULL)	Dinner	2015-06-24	MP3	user_02	R2	Snack	2016-03-01	MP4	user_01	R1	Snack	2015-06-24	MP5	user_02	R2	Lunch	2016-03-01
Recipe_ID	User_Name	Recipe_Name	Recipe_Ima																																									
R1	user_01	R1 Name	0x89504E4																																									
R2	user_02	R2 Name	(NULL)																																									
Meal_Plan_ID	User_Name	Recipe_ID	Meal_Type	Date																																								
MP1	user_01	R1	Breakfast	2015-06-24																																								
MP2	user_01	(NULL)	Dinner	2015-06-24																																								
MP3	user_02	R2	Snack	2016-03-01																																								
MP4	user_01	R1	Snack	2015-06-24																																								
MP5	user_02	R2	Lunch	2016-03-01																																								

3.7 Grocery Lists

SQL Implementation:

```
CREATE TABLE `grocery_lists` (
  `Grocery_List_ID` VARCHAR(25) NOT NULL,
  `User_Name` VARCHAR(50) NOT NULL,
  `Grocery_List_Name` VARCHAR(50) NOT NULL,
  `Grocery_List_Save_Date` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`Grocery_List_ID`),
  INDEX `FK_users` (`User_Name`),
  CONSTRAINT `FK_users` FOREIGN KEY (`User_Name`) REFERENCES `users` (`User_Name`) ON UPDATE CASCADE
)
```

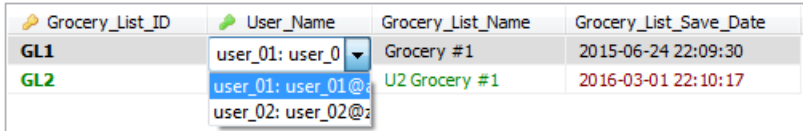
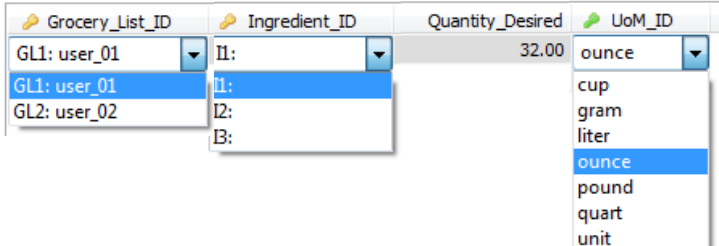
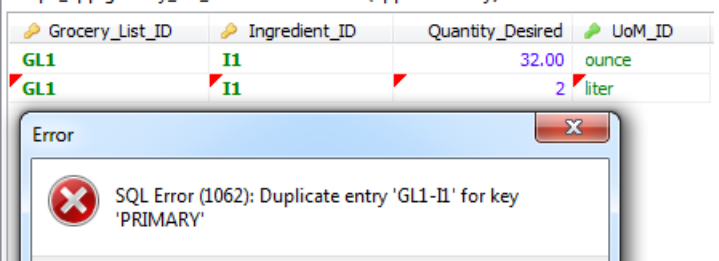
```
CREATE TABLE `grocery_list_items` (
  `Grocery_List_ID` VARCHAR(25) NOT NULL,
  `Ingredient_ID` VARCHAR(25) NOT NULL,
  `Quantity_Desired` FLOAT(25,2) UNSIGNED NOT NULL DEFAULT '1.00',
  `UoM_ID` VARCHAR(25) NOT NULL DEFAULT 'unit',
  PRIMARY KEY (`Grocery_List_ID`, `Ingredient_ID`),
  INDEX `FK_grocery_list_items_ingredients` (`Ingredient_ID`),
  INDEX `FK_grocery_list_items_measurement_units` (`UoM_ID`),
)
```

```

CONSTRAINT `FK_grocery_list_items_grocery_lists` FOREIGN KEY (`Grocery_List_ID`) REFERENCES `grocery_lists`
(`Grocery_List_ID`) ON UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT `FK_grocery_list_items_ingredients` FOREIGN KEY (`Ingredient_ID`) REFERENCES `ingredients`
(`Ingredient_ID`),
CONSTRAINT `FK_grocery_list_items_measurement_units` FOREIGN KEY (`UoM_ID`) REFERENCES `measurement_units`
(`UoM_ID`) ON UPDATE CASCADE
)

```

Tests:

Action	Expected	Actual
Add a new record to the Grocery Lists table for both users.	The User Name field is limited to the existing records in the Users table; the record is successfully added to the Grocery Lists table.	<p>recipe_app.grocery_lists: 2 rows total (approximately)</p> 
For the first user, add a new record to the Grocery List Items table associated with the created grocery list and the water ingredient (I1)	The Grocery List ID field is limited to the existing records in the Grocery List table; the Ingredient ID field is limited to the existing records in the Ingredients table; the UoM ID field is limited to the existing values in the Measurement Units validation table; the record is successfully added to the Grocery List Items table.	<p>recipe_app.grocery_list_items: 2 rows total (approximately)</p> 
Create a duplicate of the created record changing only the Quantity Desired and UoM ID fields.	The record is not added due to the composite primary key constraint placed on the Grocery List ID and Ingredient ID fields.	<p>recipe_app.grocery_list_items: 1 rows total (approximately)</p> 

Action	Expected	Actual																					
Create a duplicate of the record changing only the Grocery List ID field to the second user's Grocery List reference (GL2) and the Quantity Desired and UoM ID fields.	The record is successfully added to the Grocery List Items table.	<p>recipe_app.grocery_list_items: 2 rows total (approximately)</p> <table><tr><th>Grocery_List_ID</th><th>Ingredient_ID</th><th>Quantity_Desired</th><th>UoM_ID</th></tr><tr><td>GL1</td><td>I1</td><td>32.00</td><td>ounce</td></tr><tr><td>GL2</td><td>I1</td><td>2.00</td><td>liter</td></tr></table>	Grocery_List_ID	Ingredient_ID	Quantity_Desired	UoM_ID	GL1	I1	32.00	ounce	GL2	I1	2.00	liter									
Grocery_List_ID	Ingredient_ID	Quantity_Desired	UoM_ID																				
GL1	I1	32.00	ounce																				
GL2	I1	2.00	liter																				
Delete the water ingredient reference (I1) from the Cupboard Items table, and then delete the the water ingredient reference (I1) from the Ingredients table.	The record is successfully deleted from the Cupboard Items table; the record cannot be deleted from the Ingredients table due to the restrict deletion relationship between the Ingredients and Grocery List Items tables.	<p>recipe_app.cupboard_items: 2 rows total (approximately)</p> <table><tr><th>User_Name</th><th>Ingredient_ID</th><th>UoM_ID</th><th>Quantity_Remaining</th><th>Expiration_Date</th></tr><tr><td>user_02</td><td>I3</td><td>ounce</td><td>8.00</td><td>(NULL)</td></tr><tr><td>user_01</td><td>I3</td><td>gram</td><td>285.00</td><td>2016-06-24</td></tr></table> <p>recipe_app.ingredients: 3 rows total (approximately)</p> <table><tr><th>Ingredient_ID</th><th>Ingredient_Brand</th><th>Ingredient_Name</th></tr><tr><td>I1</td><td>(NULL)</td><td>Water</td></tr></table> <p>Error</p> <p>Grid editing error</p> <p>SQL Error (1451): Cannot delete or update a parent row: a foreign key constraint fails ('recipe_app`.`grocery_list_items`, CONSTRAINT 'FK_grocery list items_ingredients' FOREIGN KEY ('Ingredient_ID') REFERENCES 'ingredients' ('Ingredient_ID'))</p>	User_Name	Ingredient_ID	UoM_ID	Quantity_Remaining	Expiration_Date	user_02	I3	ounce	8.00	(NULL)	user_01	I3	gram	285.00	2016-06-24	Ingredient_ID	Ingredient_Brand	Ingredient_Name	I1	(NULL)	Water
User_Name	Ingredient_ID	UoM_ID	Quantity_Remaining	Expiration_Date																			
user_02	I3	ounce	8.00	(NULL)																			
user_01	I3	gram	285.00	2016-06-24																			
Ingredient_ID	Ingredient_Brand	Ingredient_Name																					
I1	(NULL)	Water																					

Non-Direct Activity Report

Date	Duration (minutes)	Specific Task / Activity
16-Jun-2016	52	Work on project #3
18-Jun-2016	226	Work on project #3
19-Jun-2016	426	Work on project #3
20-Jun-2016	108	Work on project #3
21-Jun-2016	236	Work on project #3
22-Jun-2016	280	Work on project #3
23-Jun-2016	184	Work on project #3
24-Jun-2016	311	Work on project #3
Sum for Report #1	2063	/ 1800 (2 weeks @ 900/wk)
Sum for Report #2	2313	/ 900 (1 weeks @ 900/wk)
Sum for Report #3	1823	/ 1800 (2 weeks @ 900/wk)
Sum for Class	6199	/ 4500 (5 weeks @ 900/wk)