

SSE 691

**Database Design in Software
Engineering**

Project #1

by

Jason Payne

June 3, 2016

TABLE OF CONTENTS

1. Project Overview	4
1.1 Design Process.....	4
1.2 Problem Statement	5
2. Database Mission	6
2.1 Mission Statement	6
2.1.1 Interview Questions.....	6
2.1.2 Declaring the Mission Statement	7
2.2 Mission Objectives.....	7
2.2.1 Interview Questions.....	7
2.2.2 Declaring the Mission Objectives.....	8
3. Analyzing the Existing Database.....	9
3.1 Analyzing Data Collection & Presentation.....	9
3.2 Identifying Subjects & Characteristics	13
3.3 Creating the Subject & Field Lists	15
4. MariaDB Installation & Demonstration	16
4.1 Pre-Installation Setup.....	16
4.2 Installation	18
4.3 Verification & Demonstration	21
Non-Direct Activity Report	24

Topics Covered	Topic Examples
Relational Database Design	<ul style="list-style-type: none">• Overview• Design Process
MariaDB	<ul style="list-style-type: none">• Installation• Demonstration

1. Project Overview

This project will serve as the first part in a three-part series that demonstrates the relational database design process as described in *Database Design for Mere Mortals: A Hands-on Guide to Relational Database Design, Third Edition (Hernandez, 2013)*. The next section will provide a brief summary of the overall design process that will be used. After the summary, the focus of the project will center on the design of an actual relational database for use in a potential application that is intended for those who desire an efficient way to manage recipes and menus.

1.1 Design Process

The following steps are a brief summary of the different phases of the relational database design process that will be demonstrated throughout the entirety of this project. Per the scope definitions for this project, phase 1 (defining the database's mission) and phase 2 (database analysis) will be explained in greater detail in subsequent sections.

Phases of the design process:

1. **Define a mission statement and mission objectives for the database.** The mission statement defines the overall goal and purpose of the database. The mission objectives are typically multiple statements representing the task specifications to be performed by users against the data in the database.
2. **Analyze the current database.** This phase reviews the existing processes that are currently used to collect and present data. This typically involves interviews with users and management to determine how the existing database is used on a daily basis.
3. **Create the data structures.** Tables are the first structures defined in the database and these are created by “objectifying” (in the form of tables) the subjects of the mission objectives defined in the first step.
4. **Determine and establish table relationships.** Once the tables and their associated fields have been established, the relationships between those tables are defined. This typically involves more interviews with users and management staff.
5. **Determine and define business rules.** Using feedback from users and management staff, constraints are identified and implemented in the form of business rules in order to establish various levels of data integrity.
6. **Determine and establish views.** Feedback from users and management staff are used to identify typical user interactions with data in the database. These interactions are setup as views using the appropriate tables and fields.
7. **Review data integrity.** This phase is its own process involving the following four steps:
 - a. Review the tables for compliance with proper design criteria.
 - b. Review field specifications.
 - c. Validate/test table relationships.
 - d. Review and confirm business rules.

1.2 Problem Statement

The focal point and case-study of this project will be based upon the task of creating a smart phone application and/or website that helps users manage recipes and menus along with their respective ingredient lists and cooking directions. In typical scenarios, targeted users of the app keep this information in physical forms such as cookbooks, printouts, and file folders. Also, with the numerous digital resources that provide recipes and such, it increases complexity by forcing the users to maintain physical artifacts alongside digital ones. To help alleviate these issues, the goal of the software application would be to centralize all of these artifacts into a single repository for quick and easy access by a single interface. To accomplish this, a database is desired by the creator of the app for storing and managing the numerous artifacts and its associated data. For the purpose of these projects, the software application will simply be referred to as the 'Recipe App' and to remain within scope; the focus will remain solely on the design and development of the database and not the application itself (except as necessary for the database design).

2. Database Mission

The first step in designing a database is to define the overall goal, or mission, of the database. This overall goal is known as the mission statement and is a single, concise statement that declares the purpose of the database. At a lower level, mission objectives are multiple statements that define and declare the tasks that users can perform against the data in the database. Taken together, the mission statement and mission objectives provide the focus and direction for developing a database. They also help ensure that your final database structure supports the stated purpose and tasks. This section will define the mission statement and mission objectives for the Recipe App database.

2.1 Mission Statement

For the context of this project, the mission statement will be defined by the database developer and the main developer of the Recipe App. This dialogue will take place in the form of an interview involving open-ended questions related to the ultimate purpose of the Recipe App. Once the interview is complete, the mission statement can be derived from the answers provided by the developer. This is not a comprehensive list of all of the questions that could be asked, but these questions can typically be applied in other scenarios when creating mission statements for databases.

2.1.1 Interview Questions

The interview questions and answers listed below are what were used to define the mission statement which is stated at the end of the questions.

➤ How would you describe the purpose of your application to a new user?

The Recipe App can be purposed as a convenient and efficient mechanism for managing recipes, menus, and meal plans for household kitchens and professional restaurants alike.

➤ What would you say is the purpose of your application?

The purpose of the Recipe App is to serve as a single access point to features that allow users to efficiently manage and access recipes from any source (i.e. cookbooks, websites, etc.) for the purposes of creating menus and meal plans and tracking ingredient inventory.

➤ What is the major function of your application?

The major function of the Recipe App is the ability to efficiently manage and access recipes, menus, inventory, and grocery lists all from a single access point. Once the recipe is added to the app, it can be efficiently managed, added to menus and/or grocery lists, and shared on social media websites.

➤ How would you describe what your application does?

The Recipe App makes it easier to manage the tasks involved with managing, accessing and sharing recipes; creating menus and meal plans; and determining what ingredients to buy.

➤ How would you define the single most important reason for the existence of your application?

The Recipe App exists to reduce the dependence on cookbooks and/or websites when searching for a favorite recipe. The app also helps users to reduce the time it takes to generate menus and grocery lists.

➤ Can you tell me why you believe your application needs a database?

Given the features provided by the app and the amount of data that will need to be stored with the app, there needs to be an efficient way to manage and access the recipes and its associated data. The data also needs to remain persistent across sessions.

2.1.2 Declaring the Mission Statement

Using the answers from the preceding interview, the mission statement for the database can be defined as follows:

The purpose of the Recipe App database is to maintain the application's data, and to supply information that will be utilized by different features of the software system.

2.2 Mission Objectives

The mission objectives stated below represent single tasks that will be supported by the data that is maintained by the database. This will help to define table structures, field specifications, relationship characteristics, and views of the database. These will also help define business rules and establish data integrity. These objectives will serve as a guide during development efforts ensuring that the final database structure supports the mission statement.

Similar to the process of defining the mission statement, a round of interviews will take place between the main developer and the database developer for the purpose of identifying the general tasks that need to be supported by the data in the database. The questions are open-ended so as to promote elaborate answers that yield a good deal of detail.

2.2.1 Interview Questions

➤ What types of services will the application provide?

The Recipe App will provide an associated full-scale website that will support synchronization between the site and the app. The app will support optical character recognition (OCR) services so that recipes can be scanned in from physical artifacts. The app will also support recipe and menu sharing on popular social media websites. Future versions of the app/website will include access to a public marketplace for buying/selling/trading of recipes and menus that can automatically be merged in with existing recipe and/or menu data.

➤ What kind of data will the application work with?

The application will work with the standard types of data (character strings, real numbers, etc.) but will also work with image data and RESTful-based data such as JSON and XML files.

➤ What types of reports will the application generate?

The Recipe App can be expected to generate menus and meal plans from existing recipes. Grocery lists will be generated based on ingredient items that are missing from recipes plus whatever items are manually added by the user. Reports indicating the current status of ingredient inventory should also be expected.

➤ Can you give me an idea of the things you would like to track in the database?

The Recipe App should track the recipes, menus, ingredient inventory, grocery lists and its associated data.

➤ What about any user-specific data?

Yes, absolutely! User preferences should be tracked for synchronization purposes between the app and the website.

2.2.2 Declaring the Mission Objectives

Using the answers from the preceding interview, the mission objectives for the database can be defined as follows:

- Maintain user preferences as sensitive information.
- Maintain login information as sensitive information.
- Maintain recipe data.
- Maintain menu data.
- Maintain meal plan data.
- Track ingredient inventory.
- Maintain grocery lists.

3. Analyzing the Existing Database

Analyzing the database allows the database developer to gain a clearer understanding of any existing information requirements required by the current system or process. It can also reveal structural deficiencies that may impede future information requirements. This section demonstrates the steps involved in analyzing the current system that is used by users/clients of the Recipe App by conducting an interview session with users/clients provided by the development manager of the Recipe App. Due to scope and time limitations, only a portion of a full interview session will be presented here.

3.1 Analyzing Data Collection & Presentation

The existing database for recipes and menus are typically tracked and maintained with physical artifacts and their digital counterparts. The physical artifacts can simply be a recipe contained on an index card or on a sheet in a recipe book (see Figure 1). The digital version can be a webpage that can be easily printed and stored with the other physical artifacts (see Figure 2).

STARBUCK'S CARAMEL MACCHIATO*Caramel Macchiato*

The average coffee mug is probably equivalent to a Tall (12-oz/355 mL).

I am not following precise Starbucks standards, though if you must know it's properly one shot of espresso, about .75 oz syrup (.25 oz per 4 oz liquid capacity).

milk
espresso
vanilla syrup
your favorite caramel sauce

In a regular-sized coffee mug, add vanilla syrup and steamed milk. Top with milk foam and add freshly brewed espresso through the foam. Drizzle with caramel sauce.

ICED CARAMEL MACCHIATO

Same ingredients, plus some small ice cubes and (optional) whipped cream

In a pint glass or iced tea glass (we're assuming 16 oz), add vanilla syrup and fill about 3/5 of the way with cold milk. Add ice almost to the top and pour espresso (2 shots would be typical) over the top. If you've done it right, the espresso will mix in only about halfway down without stirring. If adding whipped cream, add it here. Drizzle the top with caramel

STARBUCK'S WHITE CHOCOLATE MOCHA

2/3-cup whole milk
6 tablespoons white chocolate chips
coffee
whipped cream

Add milk and chocolate chips to saucepan and heat until chips are melted. Meanwhile make coffee in espresso machine, pour half of milk mixture in coffee mug and fill rest of mug with coffee (saving enough room for a dollop of whipped cream). Enough for 2 mugs.

STARBUCK'S CARAMEL APPLE CIDER

1 teaspoon cinnamon syrup
1 cup apple juice
1 tablespoon whipped cream
1 teaspoon caramel topping

Blend cinnamon syrup with apple juice and heat. Top with whipped cream and caramel syrup.

Figure 1: Recipes from a recipe book

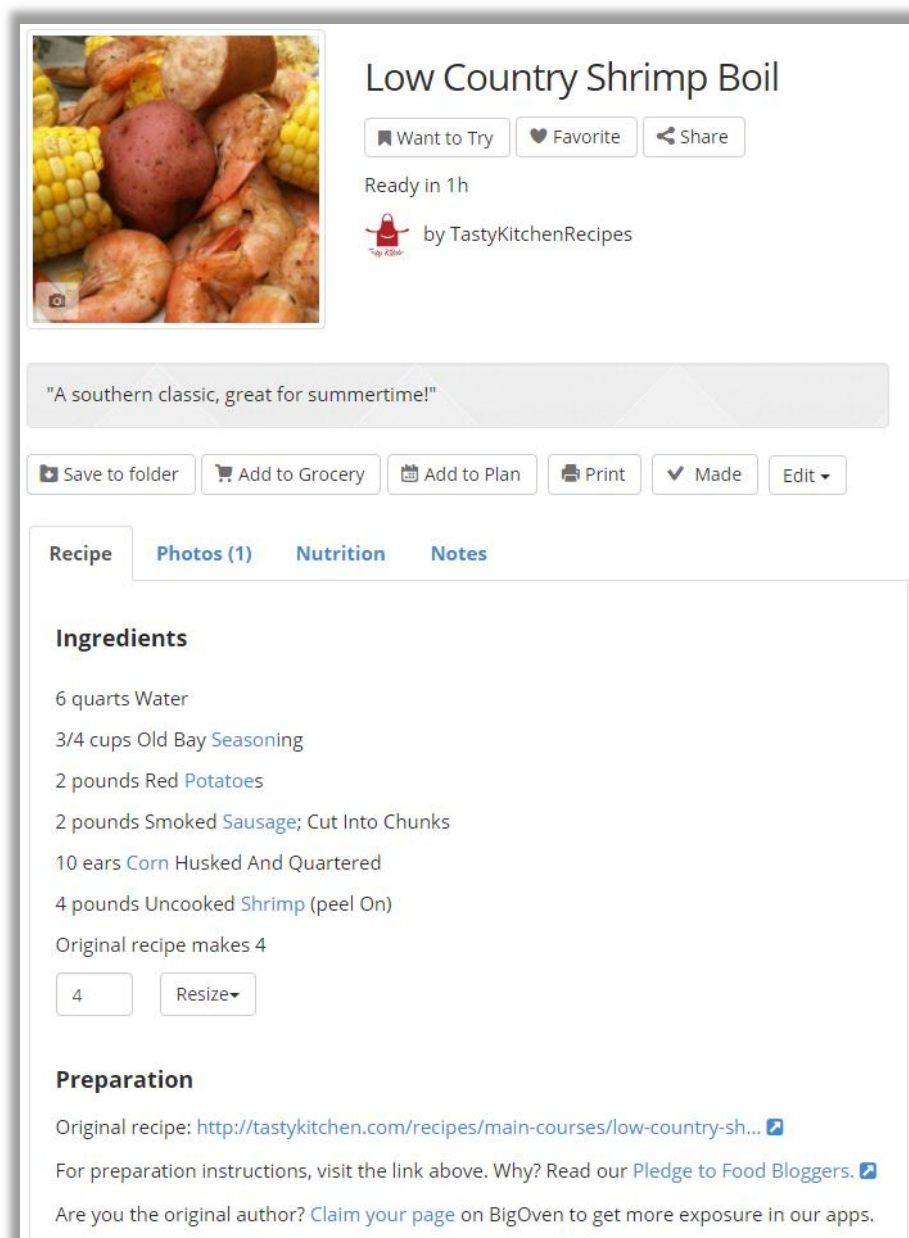


Figure 2: Digital version of a recipe

Either way, the data presented is stored in a similar format: a recipe name, an image of the prepared dish, the amount of time it takes to prepare the ingredients, the amount of time it takes to cook the recipe, the ingredients list, the preparation instructions, and the nutritional information. Cupboard inventory reports (see Table 1) and grocery lists (see Table 2) also make use of data from recipes and menus (see Figure 3).

Table 1: Hand-written cupboard inventory report

My Cupboard Inventory				
5	cans	Chicken Noodle Soup	--	Good
2	pounds	Hamburger meat	6-1-16	Expiring Soon
½	pint	1% Milk	5-15-16	Expired

Table 2: Hand-written grocery list report

Discount Foods Grocery List		
5	cans	Chicken Noodle Soup
2	pounds	Boneless chicken
1	pint	1% Milk

RIPE TOMATO CAFÉ

1410 Woodmont Ave, Bethesda MD



Appetizers

Clam Strips \$11.00
New England style, with house made sauce

🍷 **Grilled Bruschetta \$9.50**
White beans, roasted garlic, prosciutto

Antipasti \$13.00
Artisanal cheeses, prosciutto & roasted olives

Soup of the Day Cup \$4 / Bowl \$6

➔ **Peel & Eat Shrimp 1/2 lb - \$11.00**
Chilled with lemon & cocktail sauce

Smoked Salmon \$12.00
Salmon filet, toast points with house rémoulade

🍷 **Wedge Salad \$7.50**
Crumbled blue cheese, with tomato, shaved red onion & buttermilk blue dressing

➔ **Crispy Squid \$11.50**
with creole tartar

Steamed Mussels (Choice of 1) \$12.00
*Thai or *Meunière

Burgers & Sandwiches

Our Classic Burger \$7.50
Served on a fresh, brioche bun, with lettuce, tomato, onion, pickle, special sauce.

🍷 **Veggie Burger \$8.00**
Made with brown rice, black beans, oats and molasses. With lettuce, tomato, onion, pickle, and sauce.

The Southwestern Burger \$8.50
9oz burger infused with chipotles, poblanos, onion and chilis. Topped with pepper jack and served atop our black bean salsa.

Smoked Turkey Panini \$13.50
Delicate turkey breast, crispy bacon, mild cheddar cheese, vine ripened tomatoes, creamy mayo on ciabatta bread

🍷 **Veggie Medley Panini \$11.50**
Gorgonzola cheese, cucumbers, tomatoes, red peppers, and pesto spread served on ciabatta bread

🍷 SERVED WITHOUT MEAT OF ANY KIND

➔ SPECIAL THAT WE SERVE ONLY AT SPECIAL TIMES ON SPECIAL DAYS

Figure 3: Menu Example

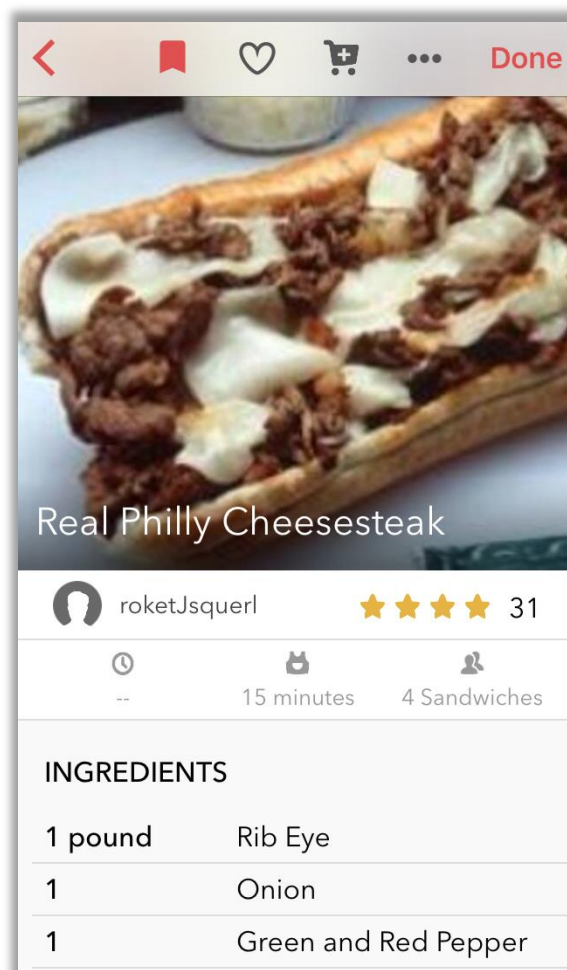


Figure 4: Recipe presented in a smartphone app

3.2 Identifying Subjects & Characteristics

The following interview illustrates a potential session conducted between the database developer and users. The goal is to help determine the subjects and characteristics that will be used in later stages of the database design. The informal subjects are highlighted in yellow and the informal characteristics are highlighted in green.

➤ How do you plan your family's eating regimen?

Person #1:

Each night, I poll my family for what they want to eat. If the results are inconclusive, then I throw something together or go get take-out. If results are more conclusive, I then check the cupboard to see if we have the proper ingredients. If we do, then I throw something together based on any instruction labels on the ingredients or I access a recipe from my recipe book. If I do not have the ingredients, then I make a quick grocery list and go to the local market to purchase the missing ingredients needed to make the recipe.

Person #2:

Each week our family decides on a **list of meals** that should be planned to be cooked for that week. I access my **recipe collection** to see if I have the proper **ingredients** for the **recipes**. If I do not have the ingredients, I create a detailed **grocery list** needed to make the recipe and proceed to the local market to purchase the missing ingredients. I then create a **schedule** specifying the day of the week that a specific recipe will be prepared.

Person #3:

I offer a **menu** of available **recipes** to my customers and each month I focus on the recipes that are being requested the most. This determines the **type** and **quantity** of **ingredients** that will be needed each week.

➤ When looking at recipes, what are the more important qualities of a recipe for you?

Person #1:

I always tend to look at how long it takes to prepare the **meal**, the **level of effort**. I then tend to look at a **picture** to see if it appears appetizing or not.

Person #2:

We like to be health conscious, so **nutritional value** is an important quality in our **meals**. **Preparation time** is also a factor, followed by **ingredient availability** and if the meal appears appetizing or not and/or if it is something we have enjoyed in the past.

Person #3:

For cost efficiency, I like to focus on whether or not I can buy the **ingredients** in bulk and how **available** (or in-season) the ingredients are at the time. I also like to focus more on offering **popular meals** that will be purchased by my customers.

➤ Using the images from Figure 1, Figure 2, and Figure 4 as a reference, what helps you prepare your meals on a day/week/month-to-day/week/month basis?

Person #1:

The **image** of the **meal** gives me a good idea if my family will like this or not. The **preparation time**, **ingredients list** (and **quantities**), and **cooking instructions** are all helpful as well.

Person #2:

Access to the **nutritional value** is really nice. Being able to adjust the **serving portion** to adjust the **ingredients' quantity** is nice too. Having easy access to the **recipe name**, **preparation time**, **ingredients list**, and **preparation steps** is also very helpful. It's nice to be able to add a recipe to a **grocery list** and/or **meal plan**.

Person #3:

Having access to the popularity rating of a recipe is very valuable to me. Being able to adjust the serving size to adjust the ingredient quantities is very helpful for me too.

3.3 Creating the Subject & Field Lists

Using the information provided from the interview session above and reviewing the screenshots of how the data is used and presented, Table 3 and Table 4 displays the subjects and preliminary/calculated fields that have been identified during the analysis, respectively.

Table 3: Subjects List

Subjects List	
• Cupboard	• Grocery Lists
• Ingredients	• Meal Plan
• Recipes	• Menus
• Ingredient List	

Table 4: Preliminary & Calculated Field Lists

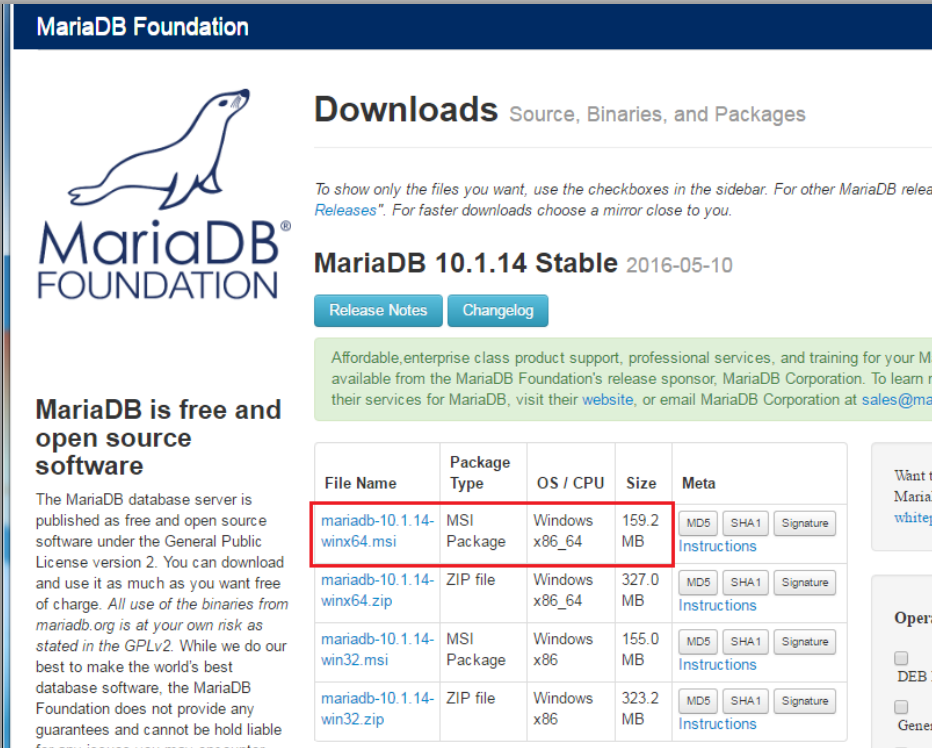
Preliminary Field List		Calculated Field List
<ul style="list-style-type: none"> • Cupboard Item • Quantity Remaining • Cupboard Item UoM • Expiration Date • Ingredient Name • Nutritional Value • Ingredient UoM • Shelf Life • Recipe Name • Picture • Popularity Rating • Preparation Time • Preparation Steps • Cooking Time • Cooking Instructions • Number of Servings • Serving UoM • Recipe Notes 	<ul style="list-style-type: none"> • Ingredient List Item • Ingredient Quantity • Ingredient List Item UoM • Grocery List Name • Grocery List Item • Quantity Desired • Quantity Desired UoM • Date • Breakfast Recipe • Lunch Recipe • Dinner Recipe • Menu Name • Menu Item • Menu Item Type (Appetizer / Entrée / Dessert / Drink) • Price • Menu Item Popularity Rating • Order Frequency 	<ul style="list-style-type: none"> • Availability • Expiration Status • Total Time • Total Nutritional Value • Total Calories
UoM = Units of Measurement		

4. MariaDB Installation & Demonstration

This section will demonstrate the procedure for installing the current stable release of MariaDB on the Windows 7 Pro operating system.

4.1 Pre-Installation Setup

The first thing to do is to get the install files for the system. These can be downloaded from the official MariaDB website: <https://downloads.mariadb.org/>. For this scenario, the 64-bit version of the MSI package will be utilized (see Figure 5).



MariaDB Foundation

Downloads Source, Binaries, and Packages

To show only the files you want, use the checkboxes in the sidebar. For other MariaDB releases, see the "Releases" section. For faster downloads choose a mirror close to you.

MariaDB 10.1.14 Stable 2016-05-10

[Release Notes](#) [Changelog](#)

Affordable, enterprise class product support, professional services, and training for your MariaDB are available from the MariaDB Foundation's release sponsor, MariaDB Corporation. To learn more about their services for MariaDB, visit their [website](#), or email MariaDB Corporation at sales@mariadb.com.

File Name	Package Type	OS / CPU	Size	Meta
mariadb-10.1.14-winx64.msi	MSI Package	Windows x86_64	159.2 MB	MD5 SHA1 Signature Instructions
mariadb-10.1.14-winx64.zip	ZIP file	Windows x86_64	327.0 MB	MD5 SHA1 Signature Instructions
mariadb-10.1.14-win32.msi	MSI Package	Windows x86	155.0 MB	MD5 SHA1 Signature Instructions
mariadb-10.1.14-win32.zip	ZIP file	Windows x86	323.2 MB	MD5 SHA1 Signature Instructions

Figure 5: MariaDB Install files from the MariaDB website

It is worth noting that MariaDB is widely used within Linux distributions and comes installed by default. However, if the Linux distribution does not have MariaDB installed, the process is still made simple with the help of the website. The website provides an interactive guide that identifies the exact procedures required to install MariaDB on specific Linux distributions. Figure 6 illustrates the steps required to install the latest stable version of MariaDB on the Ubuntu 14.04 Trusty Linux distribution.

Downloads Setting up MariaDB Repositories

To generate the entries select an item from each of the boxes below. Once an item is selected in each box, your customized repository configuration will appear below.

1. Choose a Distro

- openSUSE
- Arch Linux
- Mageia
- Fedora
- CentOS
- RedHat
- Mint
- Ubuntu**
- Debian

2. Choose a Release

- 16.04 LTS "xenial"
- 15.10 "wily"
- 14.04 LTS "trusty"**
- 12.04 LTS "precise"

3. Choose a Version

- 10.1 [Stable]**
- 10.2 [Alpha]
- 10.0 [Stable]
- 5.5 [Stable]

4. Choose a Mirror

- Accretive Networks**
- DigitalOcean - New York, US
- DigitalOcean - San Francisco, US
- James Madison University
- osuosl

[Show All Mirrors](#)

Here are the commands to run to install MariaDB on your Ubuntu system:

```
sudo apt-get install software-properties-common
sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xc9cb082a1bb943db
sudo add-apt-repository 'deb [arch=amd64,i386] http://mirrors.accretive-networks.net/mariadb/repo/10.1/ubuntu trusty main'
```

Once the key is imported and the repository added you can install MariaDB with:

```
sudo apt-get update
sudo apt-get install mariadb-server
```

See [Installing MariaDB .deb Files](#) for more information and for instructions on installing MariaDB Galera Cluster.

You can also create a custom MariaDB sources.list file. To do so, after importing the signing key as outlined above, copy and paste the following into a file under `/etc/apt/sources.list.d/` (we suggest naming the file `MariaDB.list` or something similar), or add it to the bottom of your `/etc/apt/sources.list` file.

```
# MariaDB 10.1 repository list - created 2016-05-15 03:33 UTC
# http://mariadb.org/mariadb/repositories/
deb [arch=amd64,i386] http://mirrors.accretive-networks.net/mariadb/repo/10.1/ubuntu trusty main
deb-src http://mirrors.accretive-networks.net/mariadb/repo/10.1/ubuntu trusty main
```

Figure 6: Interactive guide showing how to install stable MariaDB on Ubuntu 14.04

4.2 Installation

With the installer downloaded to the local system, the next step is to actually execute the installer wizard (see Figure 7)

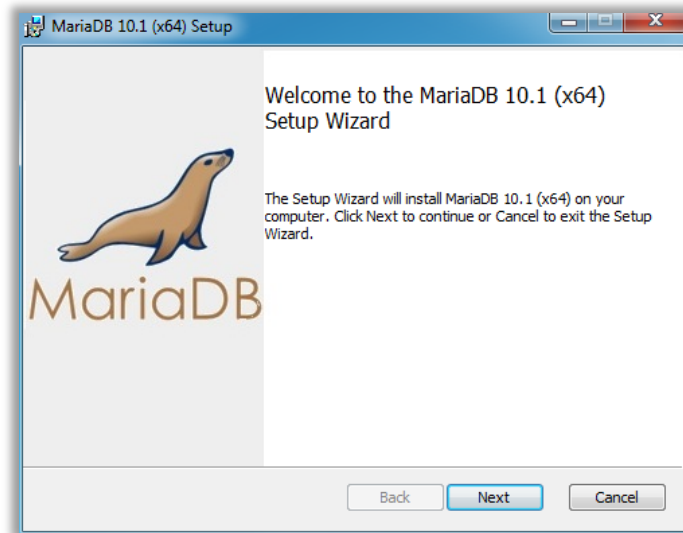


Figure 7: Welcome page of the MSI installer

Several pages will be presented by the wizard allowing the user to specify certain properties and attributes for the MariaDB instance. Figure 8 illustrates how the root password is easily established from the wizard.

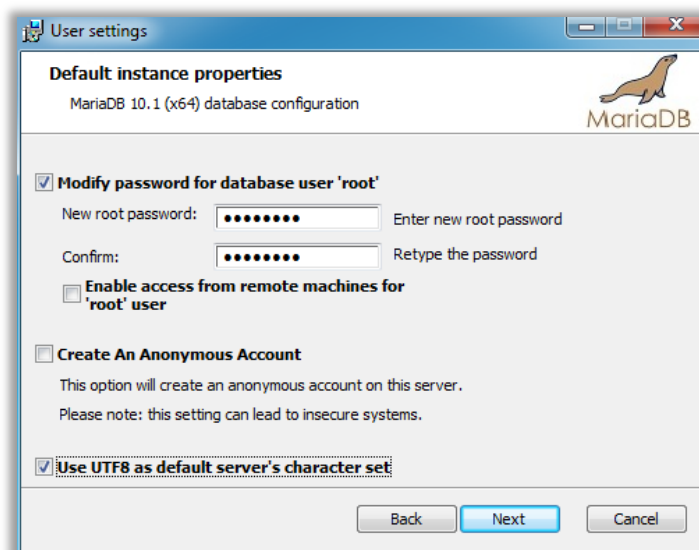


Figure 8: Specifying a root password for the MariaDB instance

For this configuration, networking was disabled to prevent remote access to the database (see Figure 9). Had this been a scenario where remote access was necessary, the 'Enable networking' checkbox would be enabled and the specific port for remote access would be specified (the default port, 3306, is acceptable in most cases though).

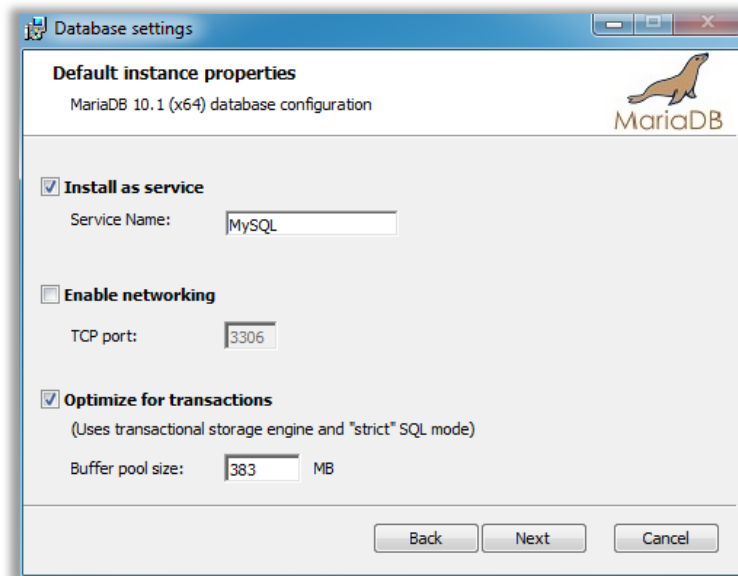


Figure 9: Specifying the MariaDB instance name

Once the properties and attributes have been specified, the database instance is ready to be installed (see Figure 10).

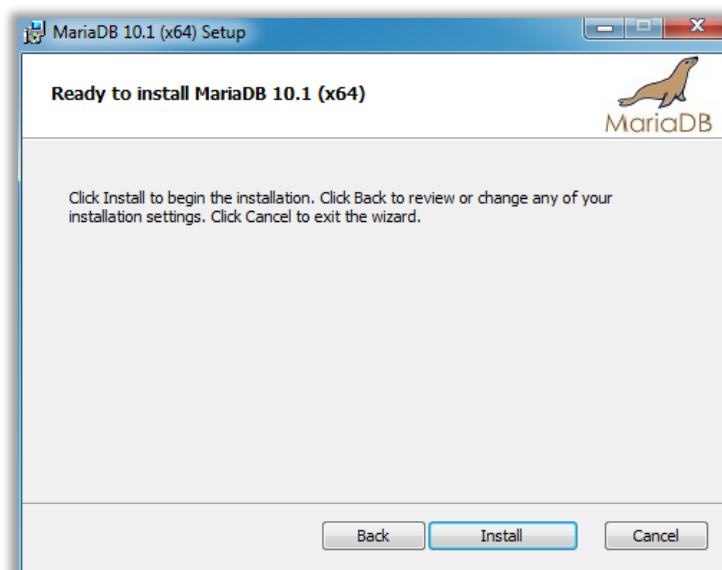


Figure 10: The MariaDB instance ready to install

Assuming that no errors are encountered during installation, the install wizard completes the installation procedure and notifies the user (see Figure 11).

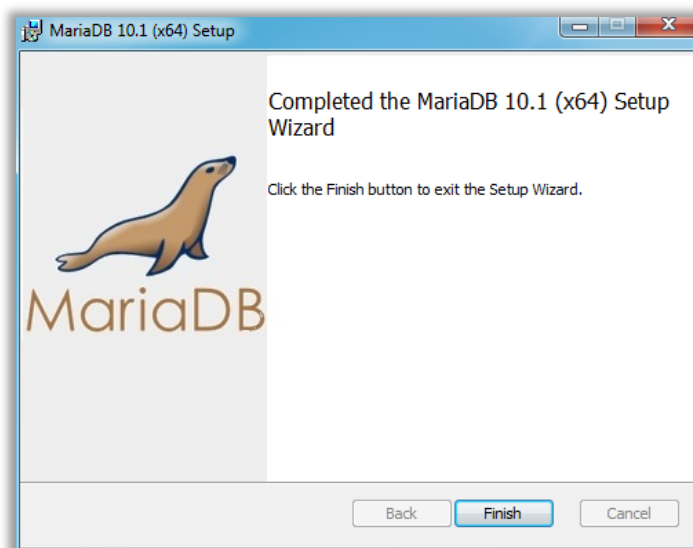


Figure 11: MariaDB install complete

With installation complete, new features and tools related to the database instance should be seen from the 'Start' menu (see Figure 12).

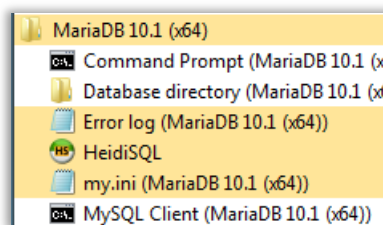


Figure 12: The new MariaDB options as seen from the Windows 'Start' menu

4.3 Verification & Demonstration

At this point, with the MariaDB instance installed, the default behavior is for the instance (or service) to automatically start running post-installation. To verify this, the MariaDB instance can be seen from the 'Component Services' application and having a service status of "Started" (see Figure 13).

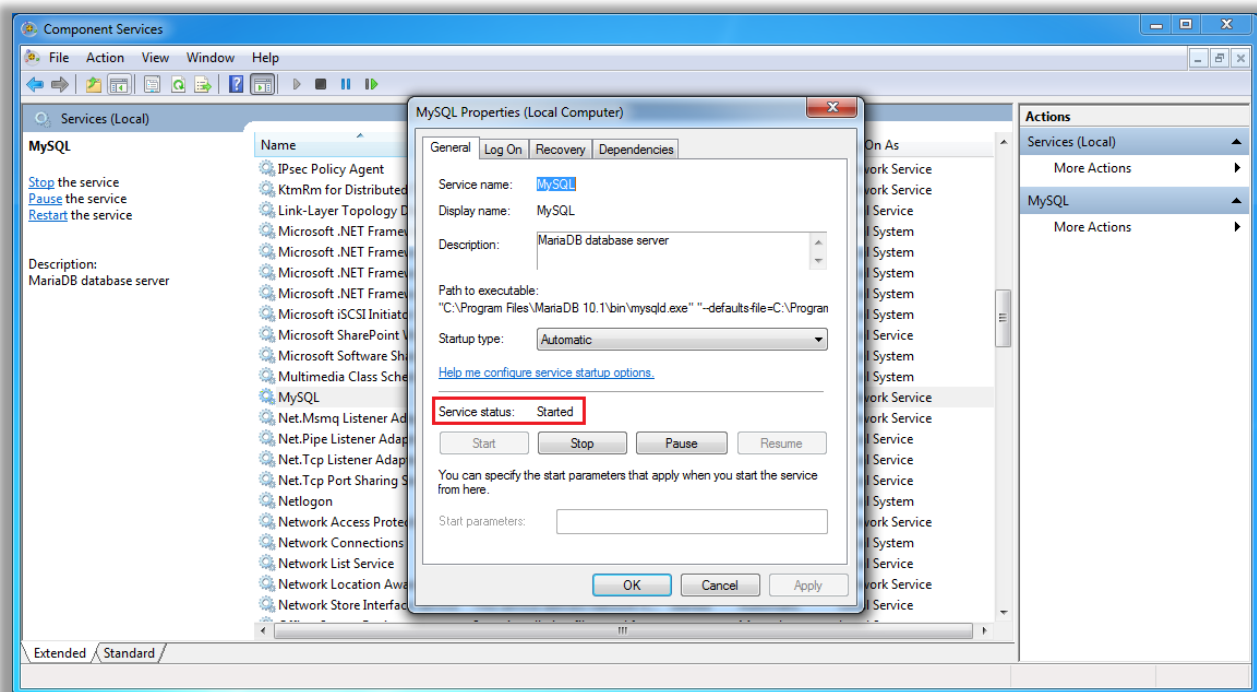


Figure 13: The running MariaDB instance on the system

The final step is to actually connect to the database in order to start designing tables and their relationships. There are two ways to do this: with the HeidiSQL application and with the MySQL Command Line Interface (CLI). Figure 14 illustrates the process of setting up a "session" that allows a user to connect to a database which is the local MariaDB in this case. For now, using the root user and the password set during installation will suffice.

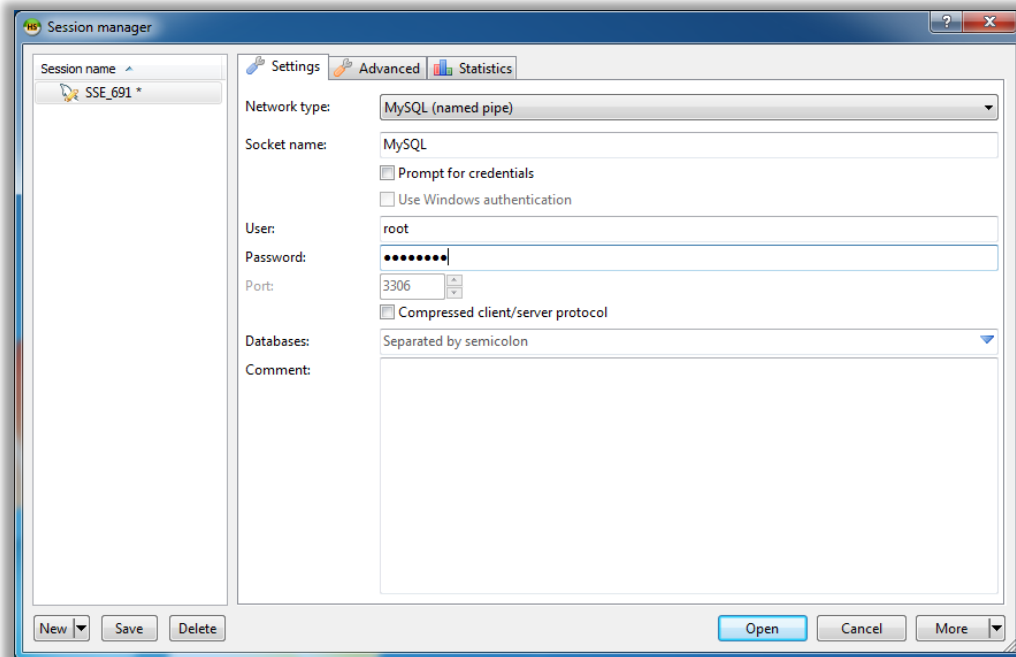


Figure 14: Logging into the MariaDB instance via the HeidiSQL application

Once connected, SQL queries can be run and tables and their relationships can be constructed (see Figure 15).

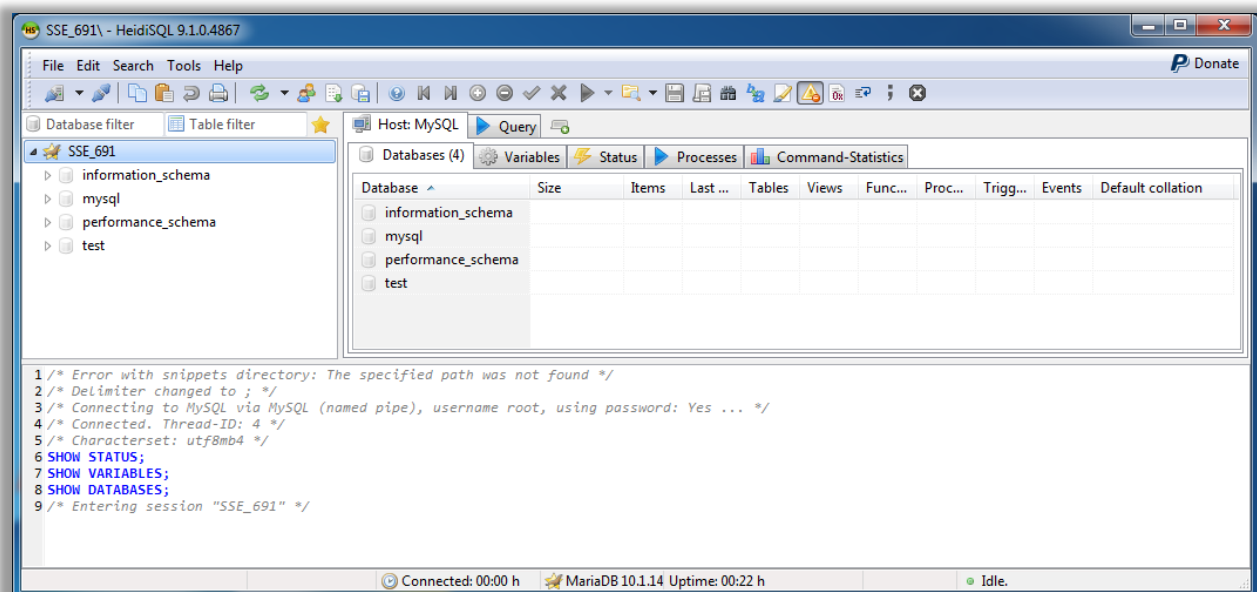
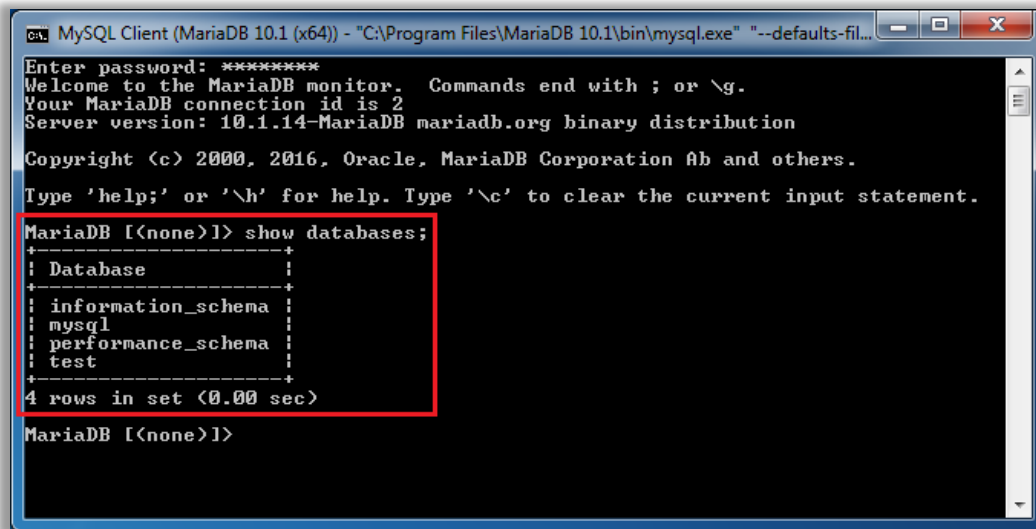


Figure 15: Connected to the local MariaDB instance through HeidiSQL

Using the MySQL CLI does essentially the same thing as the HeidiSQL application except that all interaction is accomplished with a command prompt. Figure 16 illustrates how the same information displayed by the HeidiSQL app can be shown via the `show databases;` command.



```
MySQL Client (MariaDB 10.1 (x64)) - "C:\Program Files\MariaDB 10.1\bin\mysql.exe" "--defaults-fil...
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.14-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| test |
+-----+
4 rows in set (0.00 sec)

MariaDB [(none)]>
```

Figure 16: Connected to the local MariaDB instance through the MySQL CLI

Non-Direct Activity Report

Date	Duration (minutes)	Specific Task / Activity
3-May-2016	112	Work on project #1
4-May-2016	124	Work on project #1
5-May-2016	45	Work on project #1
6-May-2016	240	Work on project #1
7-May-2016	240	Work on project #1
10-May-2016	124	Work on project #1
14-May-2016	155	Work on project #1
15-May-2016	195	Work on project #1
17-May-2016	81	Work on project #1
21-May-2016	118	Work on project #1
22-May-2016	291	Work on project #1
25-May-2016	113	Work on project #1
28-May-2016	225	Work on project #1
Sum for Report #1	2063	/ 1575 (1.75 weeks @ 900/wk)
Sum for Report #2		/ 1350 (1.5 weeks @ 900/wk)
Sum for Report #3		/ 1575 (1.75 weeks @ 900/wk)
Sum for Class	2063	/ 4500 (5 weeks @ 900/wk)