Quick Start

Installation Requirements:

Before you can successfully run my program, you must install the following program:

MySQL Installer

How to install MySQL Installer:

1.) Copy-and paste the link below into your browser:

https://dev.mysql.com/downloads/mysql/

2.) Scroll down and click under 'Select Platform' choose the operating system you are using

3.) Click to download "MySQL Installer MSI"

4.) Scroll down and click to download the "Mysql-installer-community-5.6.23.0.msi" Note- do not download "mysql-installer-web-community-5.6.23.0.msi" (choose the bottom-most Download button assuming you are using Windows)

5.) Scroll down and click "No thanks, just start my download" and Click "Save File"

6.) After the installation is complete, open the file where it was saved and run it.
7.) Click "Yes" if it asks you should it allow the mysql-installer to install
8.) Once the installer runs, Click the checkbox "I accept the license terms, followed by clicking "Next"
9.) Make sure "Developer Default" is selected and click "Next"
10.) For the "Check Requirements", click "Next" followed by "Yes" to confirm
11.) Click "Execute" to begin installing the MySQL files and Connectors
12.) When all the files completely finish installing, click "Next" to go to the "Product Configuration" setup
13.) Click "Next"
14.) In the "Type and Networking" section, make sure "Config Type" is set to "Development Machine" TCP/IP is checked, Open Firewall port for network access is checked. Also make sure the Port Number is 3306. Click "Next".
15.) Set the MySQL Root password and the Repeat password as "1234" (but exclude the quotations) then click "Next"
16.) In the Windows Service section, make sure the following has check mark selected:

-Configure MySQL Server as a Windows Service
-Start the MySQL Server at System Startup

Make sure the Windows Service Name is: MySQL56
Also make sure “Standard System Account” is selected
Click “Next”

17.) Click “Execute” to apply the changes

18.) When it is done, click “Finish”

19.) Click “Next” to configure the Samples and Examples 5.6.23

20.) Make sure the User is “root”, and the password is “1234” (excluding the quotations) then click “Check”

21.) When the connection was successful, click “Next”

22.) Click “Execute” to apply changes

23.) If they say that the configuration is taking longer than expected, click “Ok” to continue

24.) Click “Finish” when it is done to take you to the Product Configuration

25.) Click “Next”

26.) Uncheck “Start MySQL Workbench after Setup so it doesn’t run and click “Finish” to complete the installation.

How to Access MySQL

To access the MySQL on the command line:

1.) Click the Start Menu and type “cmd” (excluding the quotations) into the search bar and press “Enter” on the keyboard. (Note- If you are running Windows 8/8.1, click the Windows logo, then click the “Search” logo (the magnifying glass on the top right of the screen) and type cmd, then click “Command Prompt”)

2.) Before you can access the ledger, you must open MySQL by accessing its bin folder. MySQL is usually downloaded in the following link:

C:\Program Files\MySQL\MySQL Server 5.6\bin

Minimize the “Command line”, open “My computer” and follow the link above to make sure the bin folder is there. If it is there then highlight the above link, Right-click and click “Copy” to copy the link.

3.) Open the Command Prompt again and type “cd “ (exclude the quotations and include a space after typing cd)

4.) Right-click inside the Command line and click “Paste” to paste the link. Then press “Enter” on the keyboard. Now the directory should change to be able to access MySql

5.) Type or copy-and-paste the following into the command line:
mysql –u root –p
6.) Press “Enter”
7.) It will tell you to enter the password. Type “1234” (without the quotations) and press “Enter” on the keyboard. If it welcomes you to the MySQL monitor and you now see “mysql>” as the directory, then you are now ready to run the program. Keep the Command line open and read the next section to access the Ledger Program.

How to Access (Open) the Ledger Program

To access the Ledger Program on the command line:

Note: Make sure that the file “Zookeeper.sql” is saved on the Desktop for easy access.

1.) Type the following into the command line:

create database Ledger;

2.) Press “Enter” in the keyboard. This will create a database called “Ledger”
3.) Type the following into the command line:

connect Ledger;

4.) Click “Enter” on the keyboard. This will connect you with the “Ledger” database.
5.) To open the file, type the following into the command line:

source C:\Users\%UserName%\Desktop\zookeeper.sql

6.) Note: In place of “%UserName%” in the above line, type the folder that will lead you to the desktop in your case. In my computer for example, it is:

C:\Users\Ebuka\Desktop\zookeeper.sql

7.) Press “Enter” and you should get lines of code saying:

Query OK, 0 rows affected

8.) To see the table, type the following into the command line:

show tables;

9.) You should see a table within the database called “list”. To open the “list” table, type the following code into the command line:

Select * from list;

10.) Now you can see the ledger file with the “id, Name, Species, and Date_added” of the ledger. To add or delete animals, continue on to next section and leave command line open.

Adding and Deleting Animals from Ledger

1.) A.)To add an animal into the “list” table, type the following code into the command line:

```
INSERT INTO list (Name, Species, Date_Added)
VALUES([‘name’], [‘species’], [‘date added’]);
```

Note: Replace [‘name’], [‘species’], and [‘date_added’] with the desired name and species you would like to add. For example, If I want to add the name “Django” where its species is a “Horse” that was added on the date “April 2nd, 2015” I would type the following:

```
INSERT INTO list (Name, Species, Date_Added)
VALUES(‘Django’, ‘Horse’, ’20150402’);
```

B.) To see the updated table, type the following into the command line and press “Enter” on the keyboard after typing it:

```
Select * from list;
```

2.) A.)To remove an animal from the “list” table, type the following code into the command line:

```
DELETE FROM list
WHERE Name= [‘Animal_Name’] AND Species= [‘Species_Name’];
```

Note: Replace [‘Animal_Name’] and [‘Species_Name’] with the desired animal that you would like to remove. If for example, I want to remove the input I just added previously (Django the Horse), I would type the following into the command line:

```
DELETE FROM list
WHERE Name= ‘Django’ AND Species= ‘Horse’;
```

B.) To see the updated table, type the following into the command line and press “Enter” on the keyboard after typing it:

```
Select * from list;
```

Note: Leave your command line running and read the next section to learn how to save your file:

How to Save your Ledger database

1.) To save your database file, first you must type the following and press “Enter” after typing it:

Quit;

2.) After that you should be back in the C:\Program files..\bin directory. Now type the following to overwrite and save over the previous Ledger database:

mysqldump –u root –p Ledger > “C:\Users\%UserName%\Desktop\zookeeper.sql”

Note: Replace ‘&UserName% with the correct folder name that takes you to the Desktop Folder after you enter the “Users” folder. In my case for example, I will type:

mysqldump –u root –p Ledger > “C:\Users\Ebuka\Desktop\zookeeper.sql”

3.) Press “Enter” on the keyboard. Then type “1234” for the password (exclude the quotations).

4.) Press “Enter”

5.) To exit from the command line, type the following:

Exit

6.) Press “Enter” on the keyboard and you will exit the application.

In just a few sentences, what programming language did you use and why?

The programming language I used was SQL. The reason why I chose SQL as my programming language was because experientially I am aware that its specialty use is for creation and maintenance of databases. This challenge requires you to create a ledger storing animal information. I also chose SQL because of its capacity to be ran in a command line with MySQL which is also a constraint that I identified from the challenge instructions.

Also tell us anything else the reviewer should know about your code:

I set the id to Primary Key and to auto-increment. This means that the ledger does not need to input this data because it will automatically input a unique number.

Coding Questions

Question 1: "How did you approach the problem?"

The first and main goal I had in solving this problem was to try to systematically break the problem down into small bites. The first task was to contemplate a programming language that will tackle this problem in an efficient manner. In this case, I chose SQL. My next goal was to identify the constraints in order to successfully solve the problem. One of the constraints the problem had was that the program must run on a command line. With that being said, I decided to use MySQL on a command line. This was to enable myself to be in the shoes of the client per say. It will allow me to give the client what he wants within the same interface that he is currently using (which is command line in this case). Given the fact that my experiential knowledge on SQL was only relatively good, my next task was to focus on how I will create the bare bones of the ledger by learning about the various statements of SQL and how I can apply them to this project. As I perused through various tutorial resources, I took notes down on what statements and syntaxes will benefit me in programming this project successfully.

Before I began to code, I performed some basic programming to ensure that the command line was working properly. I then started to code, whereby my next task was to create a database followed by a table, name the table, and input the properties and data requested by the client (which are: id, Name, and Species). If I came across any errors, I would first try and repair it on my own. For example, I would first check to make sure I didn't make any spelling errors previously that might have led to it. I would also read what the error entails and see if it's a problem such as a missed semicolon to close the statement. If I couldn't fix it on my own, I would note the Error number and research online how others successfully tackled it and apply their success to my project accordingly.

After successfully coding the table on time, I now decided to do some of the bonus material. I did this because I know that Software Engineers with experience who do Agile Methodologies are always working on their program until it is just before the deadline. They tend to focus on completing what is considered high priority first and then the lower priority part follows only after completing the core wish-list. This concludes how I approached the problem.

Question 2: "What was the most difficult aspect of the solution?"

The most difficult aspect of the solution in my opinion was to do the bonus material. The reason why this was difficult to me was because it involved editing an already made table. I did manage to include the “Date_added” to the table successfully (which is considered bonus material). I had to use the INSERT INTO statement to create a new column in my “list” table, then use the UPDATE statement to update the “Date_added” column that I created.

In addition to that, the large amount of errors that I had to overcome to get the program in working order was a challenge. Although this was so, I find that my errors usually tend to be minor: either they were spelling mistakes, or that I used “SHOW” when I was supposed to use “SELECT” in the syntax for example.

Time was another challenge in this project that made it difficult. We were only given five days to complete it. I paced myself by studying the basics of SQL on Days 1 and 2. Days 2 and 3 were reserved for creating the ledger program. I planned to ideally finish on Day 3 so I can build the instructions on Day 4 and upload my program properly onto GitHub within the same day. Although time was a factor in making the challenge difficult, it made me learn about SQL in a short amount of time albeit the quality of knowledge might not be considered advanced or intermediate level, the important thing to remember was that I successfully built the program. I welcome challenges like this because it makes me sharper, smarter, and it keeps me on my toes. This is why I love Software Engineering and programming. Nobody said life is easy, that is why I always persevere through my shortcomings because that is the only way you can succeed.