DataBase Assignment

EE 470 - Introduction to the Internet of Things- Fall 2024 / Version: V2

NAME _Juan Ali Ruiz Guzman_ DATE: 03 October 2025

This project must be done individually!

Link to this document:
https://docs.google.com/document/d/1zhLaeiWjtqfpCTeJzd_n543yl6_SxahBxWd4lHh4srY/edit?usp=sharing

Please make sure you submit your assignment individually!

DataBase Assignment

EE 470 - Introduction to the Internet of Things- Fall 2024 / Version: V2

1. Assignment Overview

The purpose of this exercise is to become familiar with MySQL database and its commands.

2. Learning Objectives

- Set Up an online web page
- Learn about SQL
- Practice SQL commands
- Create a Table for sensor data
- Using Github.

3. Required Materials & Reference Information

- Signup for https://hostinger.com
- database tutorial available on the course GitHub
- Tutorial on creating an account with hostinger.com
- You can find the longitude and latitude format here
- Good tutorial: https://www.tutorialspoint.com/mariadb/mariadb_introduction.htm

4. Experiment

This experiment has TWO parts.

Part 1:
In this part of the experiment you need to signup for an account at https://hostinger.com
This will allow you to have your personal web page and database.

Part 2:
In the second part of the experiment you will need to access the database using a PHP file and visualize the data using the Chartjs library.

DataBase Assignment

EE 470 - Introduction to the Internet of Things- Fall 2024 / Version: V2

PART 1

1- A Short Introduction to Databases

A database is a systematic or organized collection of related information that is stored in such a way that it can be easily accessed, retrieved, managed, and updated. It is where all data is stored, very much like a library that houses a wide range of books from different genres. Think of data as books.

In a database, you can organize the data in rows and columns in the form of a table. Indexing the data makes it easy to find and retrieve it again as and when required. Many websites on the World Wide Web are managed with the help of databases. To create a database so that the data is accessible to users through only one set of software programs, database handlers are used.

MySQL, SQL Server, MongoDB, Oracle Database, PostgreSQL, Informix, Sybase, etc. are all examples of different databases. These modern databases are managed by DBMS. Structured Query Language, or SQL as it is more widely known, is used to operate on the data in a database.

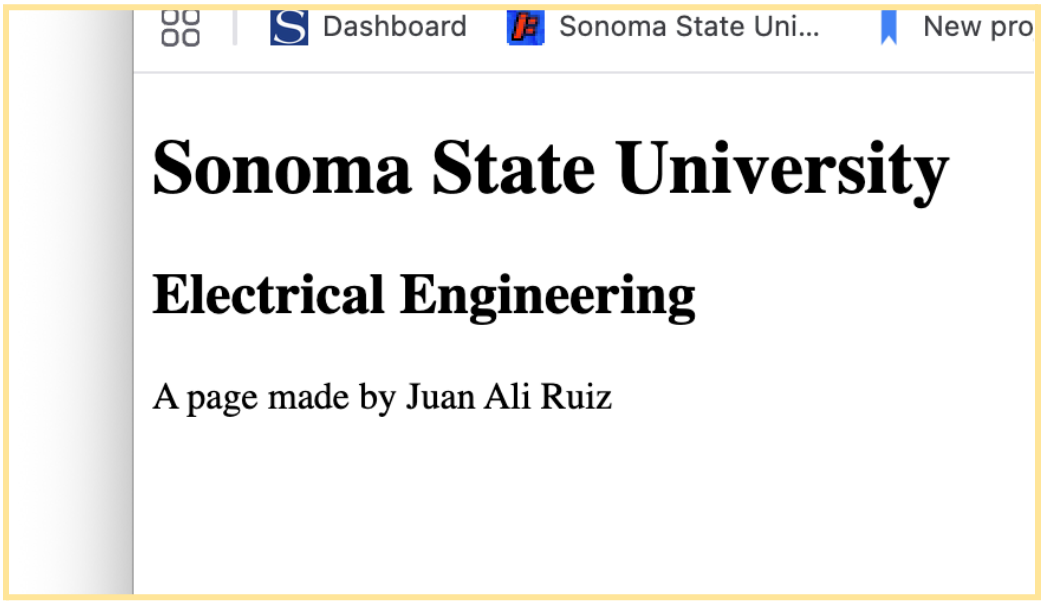
MariaDB is a popular fork of MySQL created by MySQL's original developers. It grew out of concerns related to MySQL's acquisition by Oracle. It offers support for both small data processing tasks and enterprise needs. It aims to be a drop-in replacement for MySQL requiring only a simple uninstall of MySQL and an install of MariaDB. MariaDB offers the same features of MySQL and much more.

For more information about Databases read this: https://intellipaat.com/blog/what-is-database/

1.1. Create an Account

A- Carefully review the Tutorial on creating an account with hostinger.com. Create a very simple webpage, as described in the tutorial. Make sure your web page is working. Add your name to your webpage. We will call this web page Internet of Things (IoT). We will use this web page to place links to all your future assignments.

https://juanaliruiz.com/firstpage.html
Link to your webpage



Snapshot of your webpage

B- Create a dBase. Make sure your database's name is your `YourFirstNameLastName`. The username for the database must be `db_YourFirstNameLastName`. For example database: `FaridFarahmand`
User: `db_FaridFarahmand`

You should see something like this:

Database
information_schema
u301268454_AnthonyMeyer

Follow the Tutorial and create a simple `BOOK` table. Using `SHOW DATABASES` you can show your dBase. For more information Please refer to Github Tutorial. Your final `BOOK TABLE` should look something like this:

id	name
1	MariaDB Book
2	SQLDB Book

https://github.com/faridfarahmand/SSU/blob/main/IoTCourse/dataBase/dBase_Book_Example.sql
Take a snapshot of your results:

Database: u148378080_juanaliruiz, Table: Book, Purpose: Dumping data

id	name
1	MariaDB Book
2	SQLDB Book

Snapshot of your BOOK table.

1.2. Practice with MySQL Commands

Using the phpMyAdmin in hostinger.com let's practice more with MySQL commands.

Review the commands in the database tutorial available on the course GitHub. Make sure you complete the tutorial. Once you complete the tutorial you should have a database table with your name and TWO tableshaving: `Price and Book`. Each table must have the appropriate elements.

Database: u148378080_juanaliruiz,

id	name
1	MariaDB Book1
2	MariaDB Book2
3	MariaDB Book3
4	MariaDB Book4
5	MariaDB Book5

Database: u148378080_juanaliruiz,

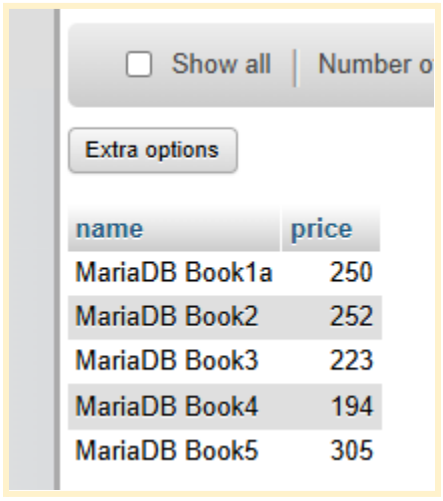
id	price
1	250
2	252
3	223
4	194
5	305

Snapshot of each table separately: `Price and Book`.

Now let's combine different fields and create ONE single table showing both book name and price - similar to below:

name	price
MariaDB Book1a	250
MariaDB Book2	252
MariaDB Book3	223
MariaDB Book4	194
MariaDB Book5	305

Take a snapshot of your results:



Snapshot: create ONE single table showing both book name and price

1.3. Create Your Own Sensory-Table

Now you need to design your own database using the following **design requirements**:

Design Two tables: `sensor_data & sensor_resister.`

`sensor_data` must have the following fields:

- node_name <up to 10 characters>; this will be used as the KEY
- time_received <indicating time/date>
- temperature <a numeric value between -10 to +100 celsius with maximum of 6 positions>
- humidity <a number value between 0-100 with maximum of 6 positions>

`sensor_register` must have the following fields:

- node_name <up to 10 characters>; this will be used as the KEY
- manufacturer <up to 10 characters>
- longitude < <a real (positive/negative number with maximum of 15 positions>
- Latitude <a real (positive/negative number with maximum of 15 positions>

You can find the longitude and latitude format here.

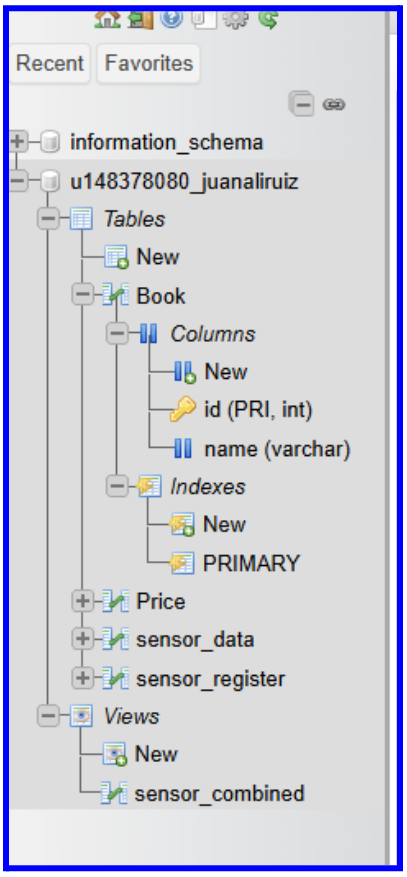
You can use a command similar to the following to create sensor data:

```
INSERT INTO sensor (node_name, time_received, temperature, humidity)
VALUES ('node-1','10-05-2022 10:05:34',84,25);
```

Your database must have the following requirements:

1- The starting data must be October 1, 2022, at 11:00 am.
2- The data must be sent every half an hour.
3- We need to register 5 different sensors with five different manufacturer names.
5- Each sensor must transmit at least 4 different data sets. This means there must be at least 4 different entries from each sensor node.
6- Node name format must be the following: node_1, node_2, node_3…. Etc.
7- No data from the sensor_data table is accepted if the transmitting node is not registered in the sensor_register table.
8- Create a VIEW of both tables and call it ensor_combined.

Once you have completed the database construction and the VIEW, take a snapshot of the content of the VIEW (sensor_combined) and all the commands you used to construct the databases. In addition, take a snapshot of the database structure, as it appears on PhpMyAdmin portal, as shown below. Your snapshots should be similar to the figures below.



Database as shown on the PhpMyAdmin page. something like the example above.

```sql
CREATE TABLE sensor_data (
    node_name VARCHAR(10) NOT NULL PRIMARY KEY,
    time_received DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    temperature DECIMAL(6,2) NOT NULL CHECK (temperature BETWEEN -10 AND 100),
    humidity DECIMAL(6,2) NOT NULL CHECK (humidity BETWEEN 0 AND 100));



CREATE TABLE sensor_register (
    node_name VARCHAR(10) NOT NULL PRIMARY KEY,
    manufacturer VARCHAR(10) NOT NULL,
    longitude DECIMAL(15,8) NOT NULL,
    latitude DECIMAL(15,8) NOT NULL,
    FOREIGN KEY (node_name)
        REFERENCES sensor_register (node_name)
        ON DELETE CASCADE
        ON UPDATE CASCADE);
```

Your complete sql script to generate the tables. Something like the example above.

Database: u148378080_juanaliruiz, Table: sensor_combined, Purpose: Dumping data

node_name	time_received	temperature	humidity	manufacturer	longitude	latitude
node_1	2022-10-01 11:00:00	22.50	45.00	Acme	-122.41940000	37.77490000
node_1	2022-10-01 11:30:00	23.50	47.00	Acme	-122.41940000	37.77490000
node_1	2022-10-01 12:00:00	24.50	49.00	Acme	-122.41940000	37.77490000
node_1	2022-10-01 12:30:00	25.50	51.00	Acme	-122.41940000	37.77490000
node_2	2022-10-01 11:00:00	22.50	45.00	Beta	-118.24370000	34.05220000
node_2	2022-10-01 11:30:00	23.50	47.00	Beta	-118.24370000	34.05220000
node_2	2022-10-01 12:00:00	24.50	49.00	Beta	-118.24370000	34.05220000
node_2	2022-10-01 12:30:00	25.50	51.00	Beta	-118.24370000	34.05220000
node_3	2022-10-01 11:00:00	22.50	45.00	Alpha	-117.16110000	32.71570000
node_3	2022-10-01 11:30:00	23.50	47.00	Alpha	-117.16110000	32.71570000
node_3	2022-10-01 12:00:00	24.50	49.00	Alpha	-117.16110000	32.71570000
node_3	2022-10-01 12:30:00	25.50	51.00	Alpha	-117.16110000	32.71570000
node_4	2022-10-01 11:00:00	22.50	45.00	Echo	-121.88630000	37.33820000
node_4	2022-10-01 11:30:00	23.50	47.00	Echo	-121.88630000	37.33820000
node_4	2022-10-01 12:00:00	24.50	49.00	Echo	-121.88630000	37.33820000
node_4	2022-10-01 12:30:00	25.50	51.00	Echo	-121.88630000	37.33820000
node_5	2022-10-01 11:00:00	22.50	45.00	Delta	-119.41790000	36.77830000
node_5	2022-10-01 11:30:00	23.50	47.00	Delta	-119.41790000	36.77830000
node_5	2022-10-01 12:00:00	24.50	49.00	Delta	-119.41790000	36.77830000
node_5	2022-10-01 12:30:00	25.50	51.00	Delta	-119.41790000	36.77830000

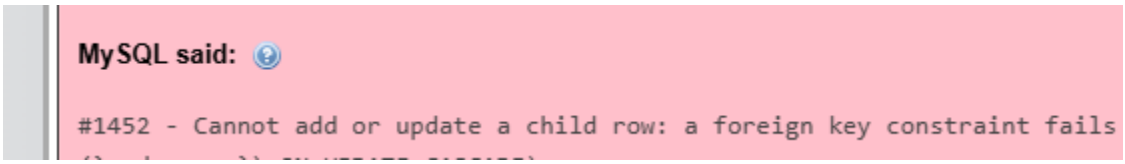
Snapshot of the VIEW (sensor_combined with all the values in there)

Prove that if you insert data for a sensor that is NOT registered the value will not be accepted.

```
198
199 INSERT INTO sensor_data (node_name, time_received, temperature, humidity)
200 VALUES ('node_unreg', '2022-10-01 11:00:00', 25.00, 50.00);
```

MySQL said:

#1452 - Cannot add or update a child row: a foreign key constraint fails (`u148378080_juanaliruiz`.`sensor_data`, CONSTRAINT `fk_sensor` FOREIGN KEY (`node_name`) REFERENCES `sensor_register` (`node_name`) ON UPDATE CASCADE)



Your snapshot

Please note: During your demonstration, you must show that all assigned features are fully functional. Each missing or non-functioning feature will result in a **7-point deduction** from your grade.

Please place your final SQL code for the sensor table in your Github. Provide the link to your Github:

https://github.com/JuanAliRuiz/EE470-Internet-of-Things/tree/main

Your Github link

Please add the link to your GitHub repository on the web page you created in **Section 1.1**.

Then, in the provided space below, provide the link to that web page. This page will serve as your **main web page for the remainder of the semester**, and all assignment links must be posted there. This is a simple template I have created: https://faridfarahmand.net/SampleStudentAssignmentIndex.html

You can copy the template page and easily update it.

https://juanaliruiz.com/

Your web page link.

2- Answer the following questions

Answer the following questions.

DataBase Assignment

1. What type of database does hostinger.com use?
   Hostinger uses a relational database.
2. What type of web server does hostinger.com use?
   Hostinger uses LiteSpeed Web Server(LSWS).
3. Referring to the data type tutorial (https://www.tutorialspoint.com/mariadb/mariadb_data_types.htm) assuming we have the following:

```
CREATE TABLE ints (a INT,b INT UNSIGNED,c INT ZEROFILL);
```

   Would this insertion command work? `INSERT INTO ints VALUES (-10,-10,-10);`
   This insert command would not work because the second value would not meet the conditions of the table. The second value inserted has to be an unsigned integer and cannot accept a negative value.
4. What is the purpose of having a primary key in a dBase?
   The purpose of having a primary key in a dBase is to uniquely identify each row in a table. This ensures data integrity and allows fast lookups and joins.
5. What is the purpose of having a foreign key in a dBase?
   The purpose of having a foreign key is to access the data recorded in one table to use in a separate table. This is great to ensure that the table data lines up with each other and creates a “parrent-child” relationship between tables.
6. Can a database have multiple tables?
   Yes, a dBase typically has many tables to better organize data.
7. What is a relational database?
   A relational database is a database that stores data in tables that can be linked to each other using keys.
8. Is MariadBase a relational or non-relational dBase?
   MariaDB is a relational database because it uses tables that are linked to each other to store data. It is a branch of MySQL and follows the relational model.
9. What is a database schema?
   The database schema is the format and overall structure of the database. It defines tables, data, and defines constraints like values and keys.
10. What is a Table in a relational Database?
    A table is a structured collection of data organized into rows and columns.
11. What are the main components of a Table in a relational database?
    The main components of a table are he rows, columns, and constraints.
12. In terms of data type, what is the difference between char(size) and varchar(size)?
    The main difference between char(size) and varchar(size) is that char is sensitive to and uses spaces to fill its size, while varchar ignores unused space and only takes in the characters.
13. What would be an example of datetime datatype format?
    The format is ‘YYY-MM-DD HH:MM:SS’, so an example would be ‘2025-10-12 15:30:00’.

END OF PART 1