User Manual

CMPSC 431W

INTRODUCTION:

This application could be an E-commerce Sales and Customer Management System designed to handle online product sales, customer orders, reviews, seller management, and payment processes. It could also include features for managing sales leads and business segments, making it suitable for a Business-to-Consumer (B2C) or Business-to-Business (B2B) platform.

This application will support an advanced e-commerce application capable of handling product listings, orders, payments, customer feedback, seller management, and even sales lead tracking. Its comprehensive business intelligence makes it adaptable for a variety of commercial and retail needs.

DATABASE DESIGN:

TABLES:

orders (order_id, customer_id, order_status, order_purchase_timestamp, order_approved_at, order_delivered_carrier_date, order_delivered_customer_date, order_estimated_delivery_date)

order_payments (order_id, payment_sequential, payment_type, payment_installments, payment_value)

customers (customer_id, customer_unique_id, customer_zip_code_prefix, customer_city, customer_state)

products (product_id, product_category_name, product_name_length, product_description_length, product_photos_qty, product_weight_g, product_length_cm, product_height_cm, product_width_cm)

order_items (order_id, order_item_id, product_id, seller_id, shipping_limit_date, price, freight_value)

order_reviews (review_id, order_id, review_score, review_creation_date)

sellers (seller_id, seller_zip_code_prefix, seller_city, seller_state)

geolocation (geolocation_zip_code_prefix, geolocation_lat, geolocation_lng, geolocation_city, geolocation_state)

SQL INJECTION PREVENT

In this project, we successfully achieved effective protection against SQL injection by using a parameterized query mechanism. This achievement significantly improves the security and reliability of the system. The specific details are as follows:

```
import mariadb

connector = mariadb.connection(user,password,.....)

cursor = connector.cursor()

cursor.execute("SELECT table1.key1 FROM table1 WHERE table1.key2 = ?;",
( filters['table1']['key2'],))

result = cursor.fetchall()

cusor.execute("INSERT INTO table2 （key1, key2, key3) VALUES (?, ?, ?);",
(filters['table2']['key1'], filters['table2']['key2'], filters['table2']['key2']);
```

Parameterized queries effectively prevent SQL injection attacks, such as malicious users entering "1 OR 1=1" and other operations that attempt to bypass validation.

Our database system can always distinguish between user input and SQL commands during operation, fundamentally eliminating potential attacks. The risk of data being maliciously tampered with or accidentally leaked is avoided, especially when dealing with sensitive data such as user passwords or transaction records.

SYSTEM REQUIREMENT:

1. Install MariaDB https://mariadb.org/download/

   Here is the link to the tutorial of how you should download Maria DB and create users https://www.mariadbtutorial.com/getting-started/install-mariadb/

2. Install Python https://www.python.org/downloads/

3. Inside the python shell, make sure you download the following tools for this project:

   mariadb streamlit cryptography pyyaml

   install them separately use “pip install module_name”

4. Open Maria DB shell and create users for this project as following:

```
-- Drop Current user
```

```
DROP SER 'user_name'@'localhost';

-- OR

DROP SER 'user_name'@'%';

-- Drop User Validation

SELECT User, Host FROM mysql.user;

--Create user with hash password

SELECT PASSWORD('user_password') AS hashed_password;

-- You get a string like *7FA3F75E87E0524F77B9F0D00BF78B8E61D4A4BE , I call it
user_hashed_password

-- Create user

CREATE USER 'user_name'@'%' IDENTIFIED BY PASSWORD 'user_hashed_password';

-- OR

CREATE USER 'user_name'@'localhost' IDENTIFIED BY PASSWORD
'user_hashed_password';

-- command to log in:

mariadb -u user_name -p

--then you just input your password and then access to your database!
```

Notes: do not forget your username and password!

5. Import data into the database:

- Log in to your Marai DB
- Make sure you download the data "backup.sql" or contain it inside the folder
- Run the following command:

```
CREATE DATABASE your_database;

USE your_database;

source backup.sql
```

6. Secret yourself!

Before you run the streamlit for the last step, you need to secret yourself from the files.

a. open up the file “secret.sample.yaml”

b. copy and paste the username, user password and the name of database you created from step 4 into the corresponding field in this file

c. change the name of this file from “secret.sample.yaml” into “secret.yaml”

d. run “python encrypt.py”.

e. Don’t forget to enter your password when you finally access the web!

7. Run it!

After all the steps above, you finally reach the end!

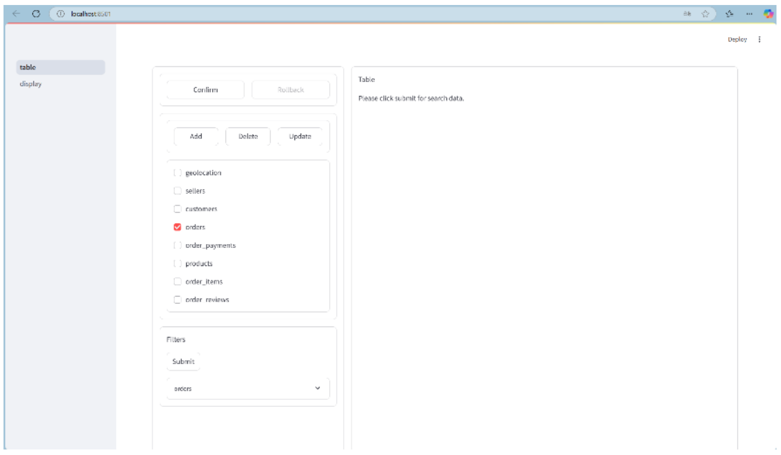
Run “streamlit run app.py”, enter the password you generated and access the data!

FUNCTIONALITY:

Because this system contains a large amount of data, please select the tables slowly and use the corresponding applications. Let us give some time for the data to be processed.

1. When you first get into (our functionality will allow you to access it without log in) this system, the following page is what you are going to see. The “order” table is the default table when you first open the system. There is also a hint on the part of table which helps you function the system by clicking the “Submit” button.

   You can select multiple tables from the checkboxes on the left to show the combined data table (slowly, please allow the system process for large amount of data).



Please click submit for search data.

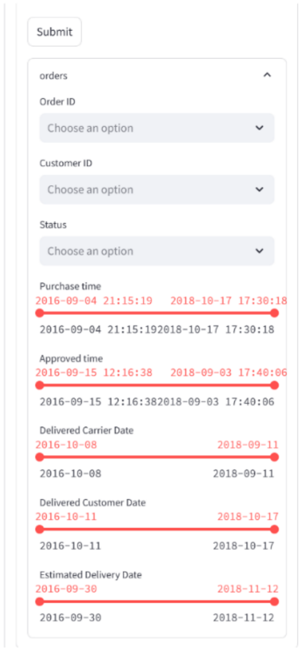
2. Accurate Filters

For the filters on the left side, when you expand the drop-down box of tables you chose, you will see the attributes for the selected table to have an accurate search.

Each drop-down box has a selection function. There is also a function to search by first letter or similar letters, providing users with a faster and more convenient search.

The moving block below will facilitate users to perform range search. However, when the red dots overlap, accurate search can also be achieved.

When the user selects multiple tables, there will be multiple filtering options corresponding to different tables.


Submit

orders

Order ID: Choose an option

Customer ID: Choose an option

Status: Choose an option

Purchase time: 2016-09-04 21:15:19 – 2018-10-17 17:30:18

Approved time: 2016-09-15 12:16:38 – 2018-09-03 17:40:06

Delivered Carrier Date: 2016-10-08 – 2018-09-11

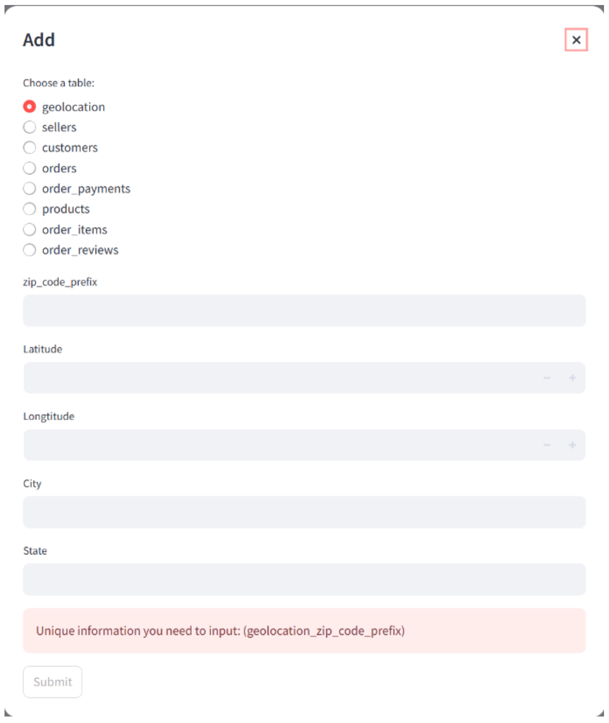
Delivered Customer Date: 2016-10-11 – 2018-10-17

Estimated Delivery Date: 2016-09-30 – 2018-11-12

3. The system can implement the functions of adding.

Users can implement these functions through the three buttons on the left side of the page without having to select the table to be changed in advance.

When the user clicks the "Add" button, the following interface will appear. According to the prompts on the interface, users can accurately add the content they want. The red content


Add

Choose a table:

[✓ geolocation] sellers customers orders order_payments products order_items order_reviews

zip_code_prefix

Latitude

Longtitude

City

State

Unique information you need to input: (geolocation_zip_code_prefix)

Submit

at the bottom of the pop-up page will prompt the user for the content required for the selected form.

After filling in all the content, click the "Submit" button below, then click the “Confirm” button. And the content entered by the user will be updated to the form synchronously.

If you are not sure about the content you entered, you can use “Rollback” button to undo it before you use the “Confirm” button.

4. <u>The system can implement the deletion function.</u>

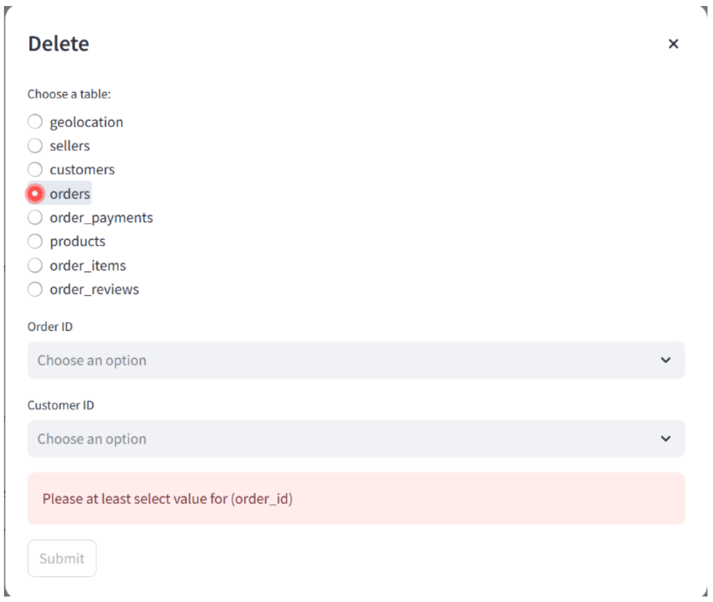
When the user clicks the "Delete" button, the following interface will appear.

According to the prompts on the interface, users can delete the desired content according to the selected table. It should be noted that users only need to select the corresponding primary key and unique key to delete the corresponding content.

The red content at the bottom of the pop-up page will prompt the user with the necessary information for deleting the content of the selected table.

After filling in all the content, click the "submit" button below, then click the “Confirm” button. And the content entered by the user will be deleted synchronously in the table.

If you are not sure about the content you entered, you can use “Rollback” button to undo it before you use the “Confirm” button.



Delete

Choose a table:

- geolocation
- sellers
- customers
- [✓ orders]
- order_payments
- products
- order_items
- order_reviews

Order ID

Choose an option

Customer ID

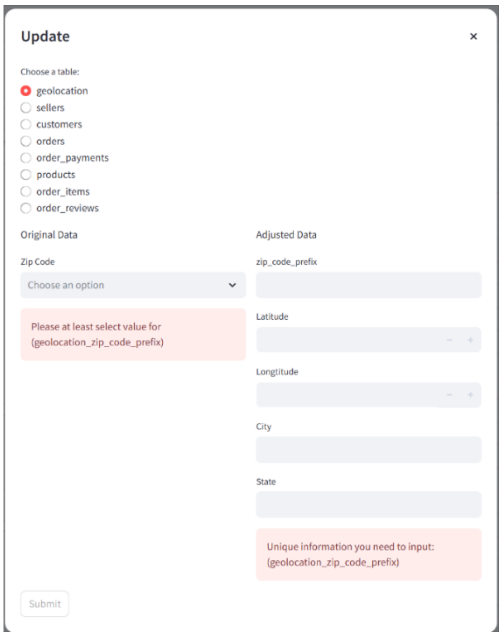
Choose an option

Please at least select value for (order_id)

5. The system can implement the update function.

When the user clicks the "Update" button, the following interface will appear.

According to the prompts on the interface, users can update the desired content according to the selected table. It should be noted that users only need to select the corresponding primary key and unique key. After selecting the information on the left, you need to fill in the new information you want to update on the right.



Please at least select value for (geolocation_zip_code_prefix)

Unique information you need to input: (geolocation_zip_code_prefix)

The content in the red box will prompt the user with the necessary information for updating the content of the selected table.

After filling in all the content, click the "submit" button below, then click the “Confirm” button. The content entered by the user will be deleted synchronously in the table. If you are not sure about the content you entered, you can use “Rollback” button to undo it before you use the “Confirm” button.

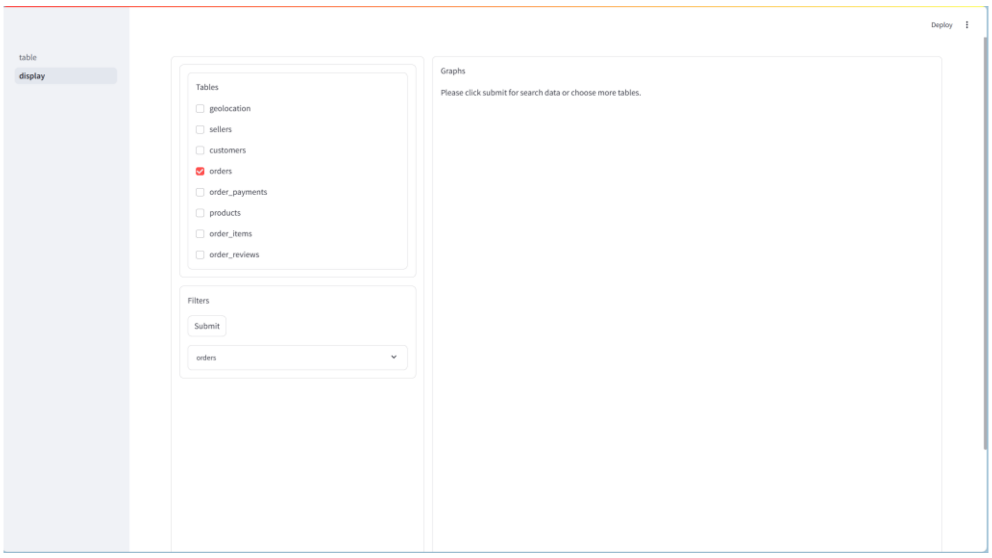
6. Data analysis and visualization

Our system supports the processing of large amounts of data and the visualization of data statistics. Users can jump to the corresponding interface by clicking the "display" button at the bottom of the leftmost sidebar.

When you click this button for the first time, you will see the following interface:

Similar to the "table" interface, users can freely check the table combination they want to analyze and obtain the corresponding image by clicking the "submit" button below.

At the same time, the display function also supports the same "filter" function as the "table" page, helping users to accurately select the desired data type or data range.

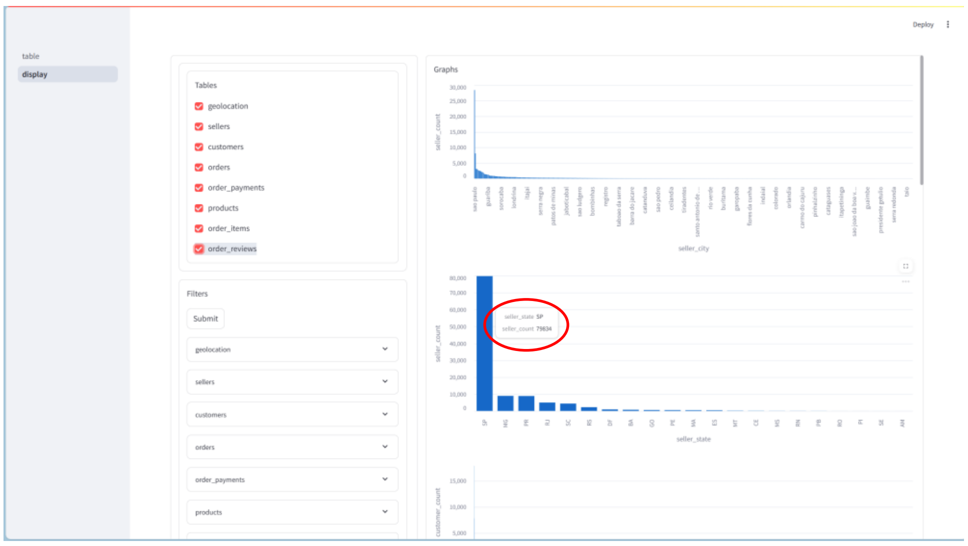


Please click submit for search data or choose more tables.

If the image is not displayed after clicking "submit", it means that this table does not have any data support for visualization. Select more tables according to the interface prompts. For example, the "geolocation" table only contains all the geographical locations that have appeared, without any course analysis data.

7. <u>Intuitive data display</u>

As shown in the figure, after the user has selected the table they want to analyze, by hovering the mouse, a small box will follow to display the specific data, making it convenient for the user to know the specific data and conduct corresponding analysis.



THANK YOU FOR USING THE SYSTEM WE BUILT!

PLEASE ENJOY YOUR DATA EXPLORATION!