Disaster Relief and Donation Management System

Project Report

Group 12

AS20240395- K I R Abeysekara

AS20240427- B P R Sachintha

AS20240404- A H M D Silva

AS20240577- N G Sachindu Sathsara

AS20240546- T H Sandaruwan

AS20240538- S M Wijesekara

AS20240559-E.K.S Govinda

CSC 1042 Database Management Systems

Lecturer in Charge: Dr. Surani Tissera

Department: Department of Computer Science

Submission Date: 01.02.2026

Table of Contents

1. Problem Statement

In the aftermath of natural disasters, the immediate management of relief efforts is often chaotic and inefficient. Manual methods of tracking donations, inventory, and distributions lead to significant data inconsistencies, loss of resources, and uneven aid distribution. There is often a lack of transparency regarding which donors have contributed, what items are currently in stock, and which beneficiaries have already received aid.

The Solution: We have designed and implemented the **Disaster Relief and Donation Management System**. This relational database system streamlines the workflow by digitalizing the tracking process. It ensures:

- **Donors** and their specific **Donations** are accurately recorded and retrievable.
- **Inventory** is automatically updated based on incoming donations and outgoing distributions.
- **Beneficiaries** are tracked to prevent duplicate distributions and ensure fair aid allocation.
- **Users** (Staff/Admins) have secure access to manage records.

2. System Scope and Limitations

This project focuses on designing and implementing a Disaster Relief and Donation Management System with primary emphasis on relational database design and SQL operations.

System Scope

The scope of the system includes:

- Designing a relational database schema using ER modeling and logical design principles.
- Defining primary keys and foreign key relationships between entities such as Donor, Donation, Inventory, Beneficiary, Distribution, and Item.
- Executing SQL DML operations (INSERT, SELECT, UPDATE, DELETE) to manage data in the database.
- Demonstrating database functionality using SQL queries executed directly in phpMyAdmin.
- Providing a basic PHP-based user interface to trigger selected SQL operations for ease of interaction and demonstration.
- Managing inventory and distribution data primarily through insert and select operations, reflecting real-world transactional behavior.

3. Conceptual Design

3.1 Assumptions

The following assumptions were made during the creation of the Entity-Relationship (ER) Diagram:

1) Each donor is uniquely identifiable

Assumption:
Each donor is uniquely identified by a system-generated DonorID.

Reason:
Donor names and contact numbers may not be unique. Therefore, a surrogate primary key is required to uniquely identify each donor.

Impact on ER Diagram:
DonorID is used as the primary key of the **Donor** entity.

2) A donor can make multiple donations

Assumption:
A single donor can make multiple donations over time, while each donation is associated with exactly one donor.

Reason:
This reflects real-world donation behavior where donors may contribute more than once.

Impact on ER Diagram:
A one-to-many relationship exists between **Donor** and **Donation**, with DonorID included as a foreign key in the **Donation** entity.

3) Inventory items are centrally managed

Assumption:
Inventory represents relief items stored and managed by the organization, and each item is uniquely identifiable.

Reason:
Centralized inventory management is required to accurately track available relief items.

Impact on ER Diagram:
The **Inventory** entity contains a unique primary key (InventoryID) and maintains item quantity information.

4) Inventory quantities change through system operations

Assumption:
Inventory quantities may change due to donations received or distributions made.

Reason:
Inventory levels must reflect real-time stock availability.

Impact on ER Diagram:
The **Inventory** entity allows controlled quantity updates through system operations.

5) Distribution records represent relief allocation events

Assumption:
Distribution records capture details of relief items distributed to beneficiaries and are mainly used for record-keeping.

Reason:
Distribution data should not be frequently modified once recorded to ensure data integrity.

Impact on ER Diagram:
The **Distribution** entity is modeled as a separate transactional entity with limited update and delete operations.

6) Each entity uses a single-attribute primary key

Assumption:
All entities use a single-column primary key.

Reason:
Single-attribute primary keys simplify database design and improve clarity.

Impact on ER Diagram:
No composite primary keys were used in the database design.

7) User authentication is outside the project scope

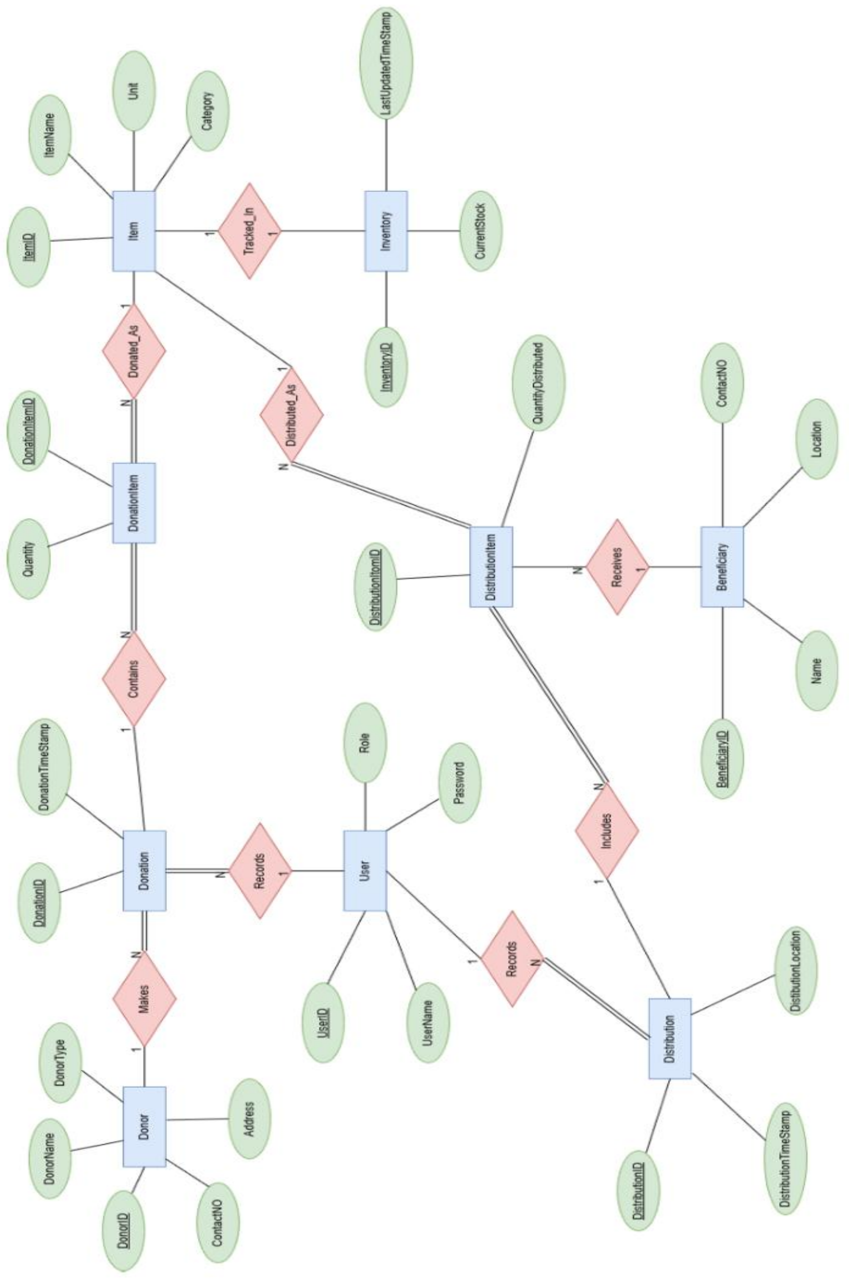
Assumption:
The system supports authenticated users with predefined roles (admin and staff), where administrative users have elevated privileges compared to staff users.

Reason:
To ensure data security and integrity, role-based access control is enforced at the application level. Editing and deletion operations are restricted to administrative users, while staff users are limited to data entry and viewing functions.

Impact on ER Diagram:
The User entity includes a role attribute to represent different user types, while permission enforcement is implemented at the application level.

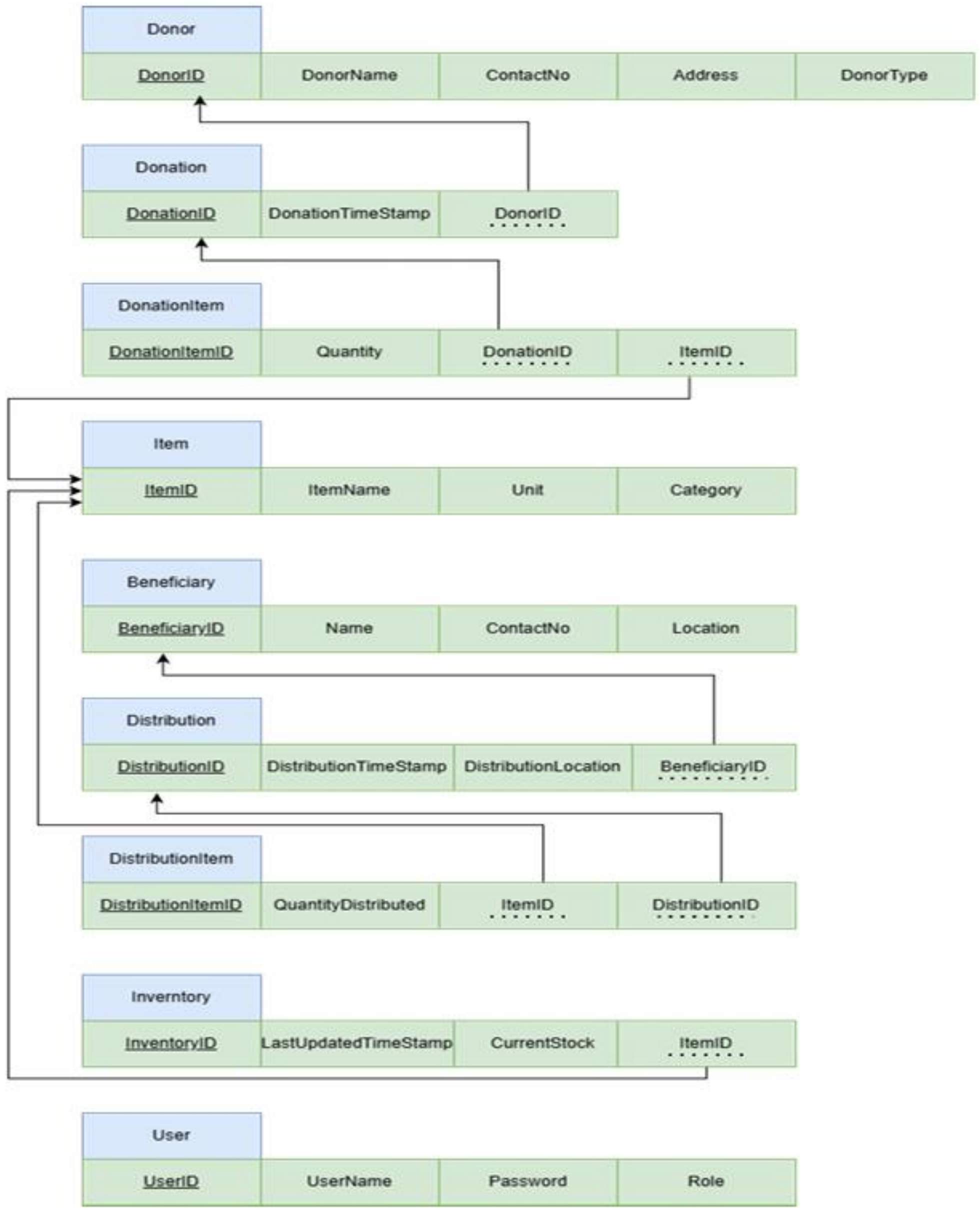
3.2 Entity-Relationship (ER) Diagram

The ER Diagram below illustrates the conceptual model of the system, defining the entities and their relationships.



4. Logical Design (Relational Schema)

The conceptual design was converted into the following Relational Schema. Primary Keys (PK) and Foreign Keys (FK) are identified to maintain referential integrity.



5. Implemented Database

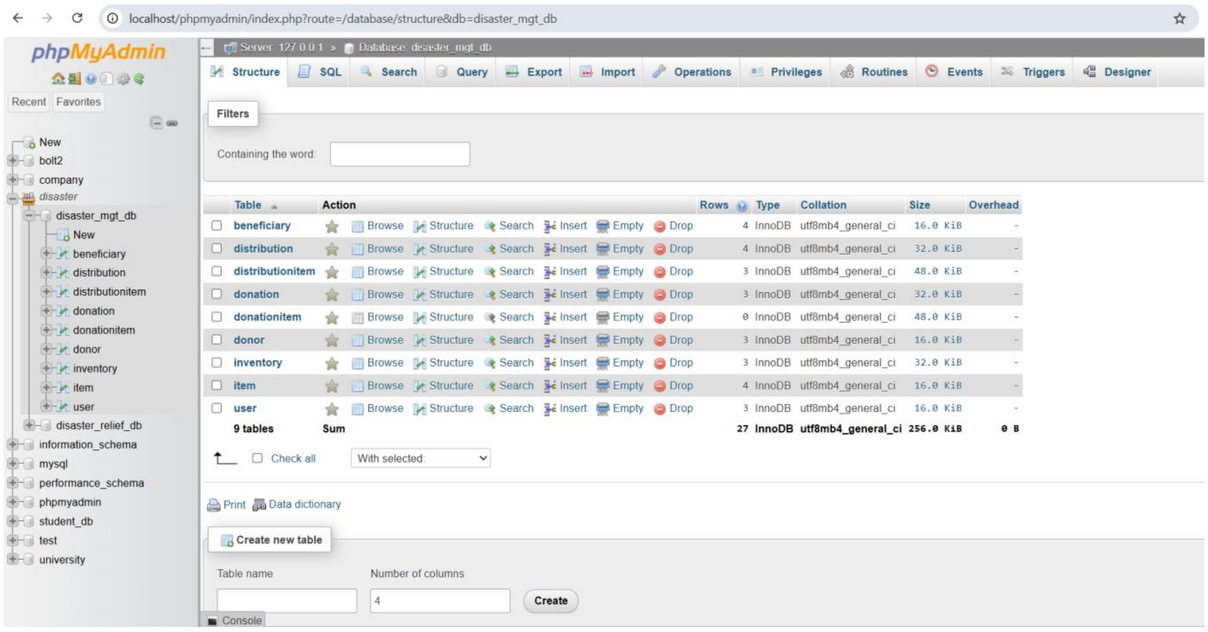


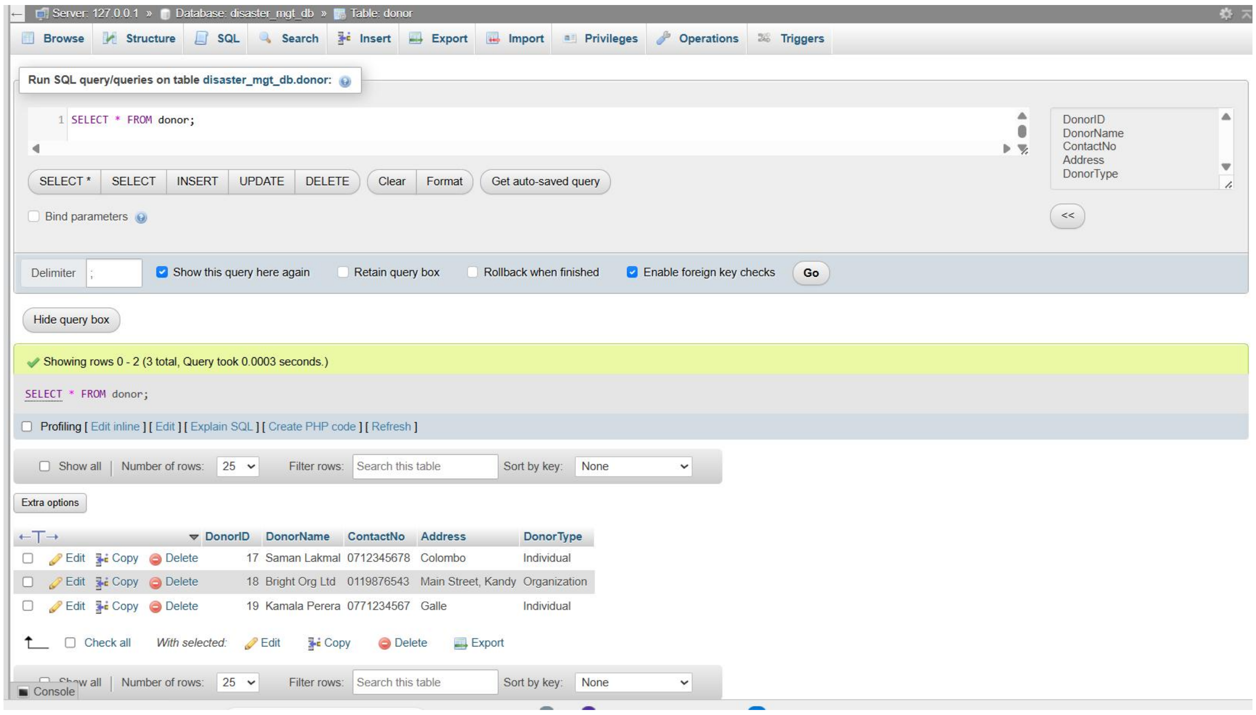
Table	Action	Rows	Type	Collation	Size	Overhead
beneficiary	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
distribution	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
distributionitem	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	48.0 KiB	-
donation	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
donationitem	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	-
donor	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
inventory	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
item	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
user	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
9 tables	Sum	27	InnoDB	utf8mb4_general_ci	256.0 KiB	0 B

Number of columns: 4

6. Populated Database Tables

The following screenshots demonstrate that the database has been successfully created and populated with data.

Table: Donor



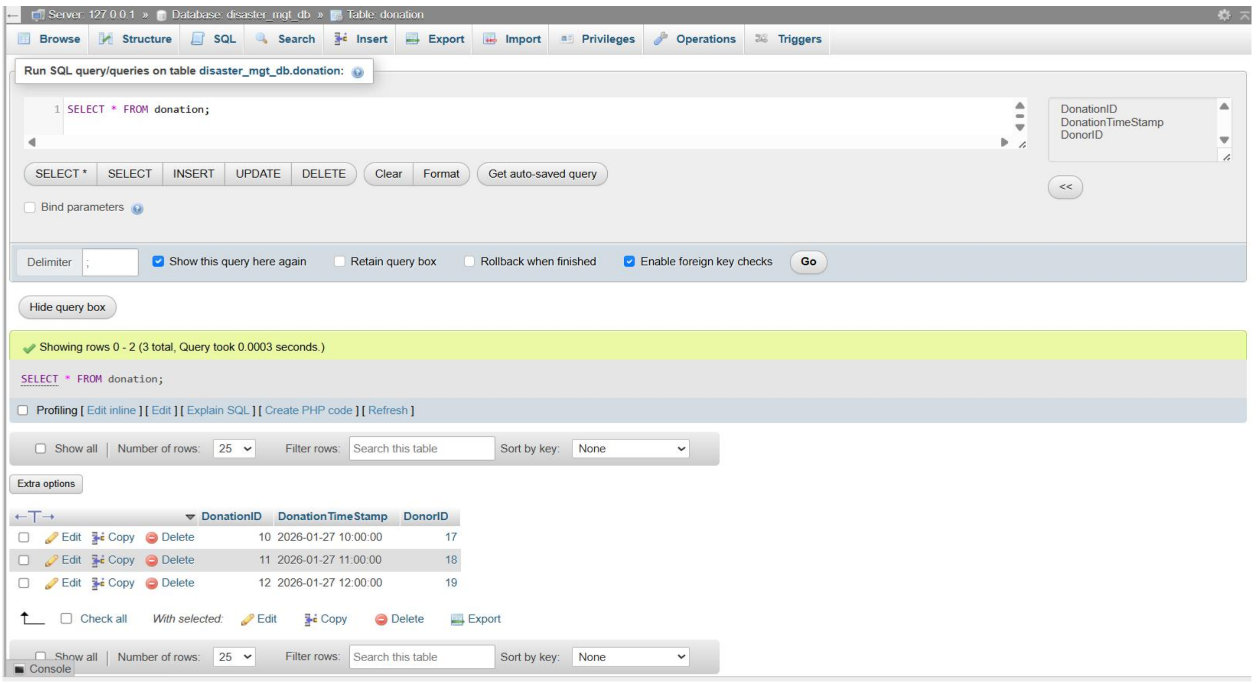
```
SELECT * FROM donor;
```

Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

```
SELECT * FROM donor;
```

	DonorID	DonorName	ContactNo	Address	DonorType
Edit Copy Delete	17	Saman Lakmal	0712345678	Colombo	Individual
Edit Copy Delete	18	Bright Org Ltd	0119876543	Main Street, Kandy	Organization
Edit Copy Delete	19	Kamala Perera	0771234567	Galle	Individual

Table: Donation



Run SQL query/queries on table disaster_mgt_db.donation:

```
SELECT * FROM donation;
```

DonationID
DonationTimeStamp
DonorID

Delimiter ; [✓ Show this query here again] Retain query box Rollback when finished [✓ Enable foreign key checks] Go

Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

```
SELECT * FROM donation;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

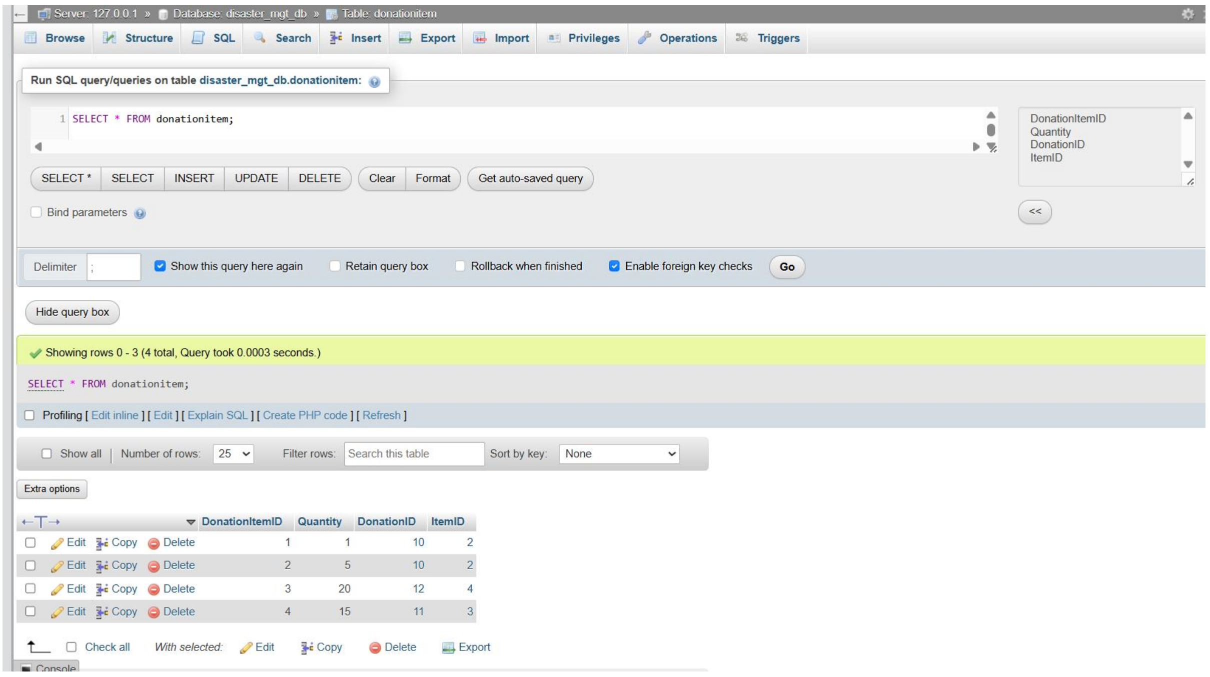
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

	DonationID	DonationTimeStamp	DonorID
Edit Copy Delete	10	2026-01-27 10:00:00	17
Edit Copy Delete	11	2026-01-27 11:00:00	18
Edit Copy Delete	12	2026-01-27 12:00:00	19

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Table: DonationItem



Run SQL query/queries on table disaster_mgt_db.donationitem:

```
SELECT * FROM donationitem;
```

DonationItemID
Quantity
DonationID
ItemID

Delimiter ; [✓ Show this query here again] Retain query box Rollback when finished [✓ Enable foreign key checks] Go

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

```
SELECT * FROM donationitem;
```

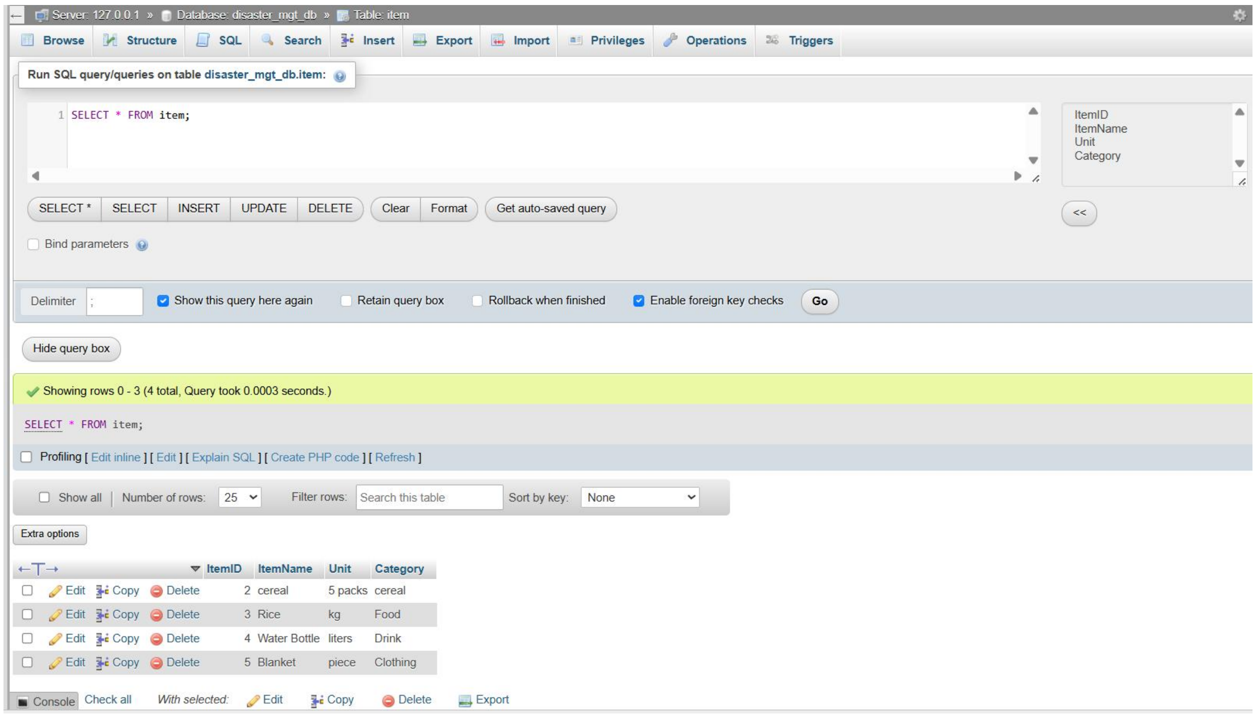
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

	DonationItemID	Quantity	DonationID	ItemID
Edit Copy Delete	1	1	10	2
Edit Copy Delete	2	5	10	2
Edit Copy Delete	3	20	12	4
Edit Copy Delete	4	15	11	3

Check all With selected: Edit Copy Delete Export

Table: Item


Server: 127.0.0.1 » Database: disaster_mgt_db » Table: item
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Run SQL query/queries on table disaster_mgt_db.item:

```
SELECT * FROM item;
```

ItemID
ItemName
Unit
Category
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
Bind parameters
Delimiter ;
Show this query here again
Retain query box
Rollback when finished
Enable foreign key checks
Go
Hide query box

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

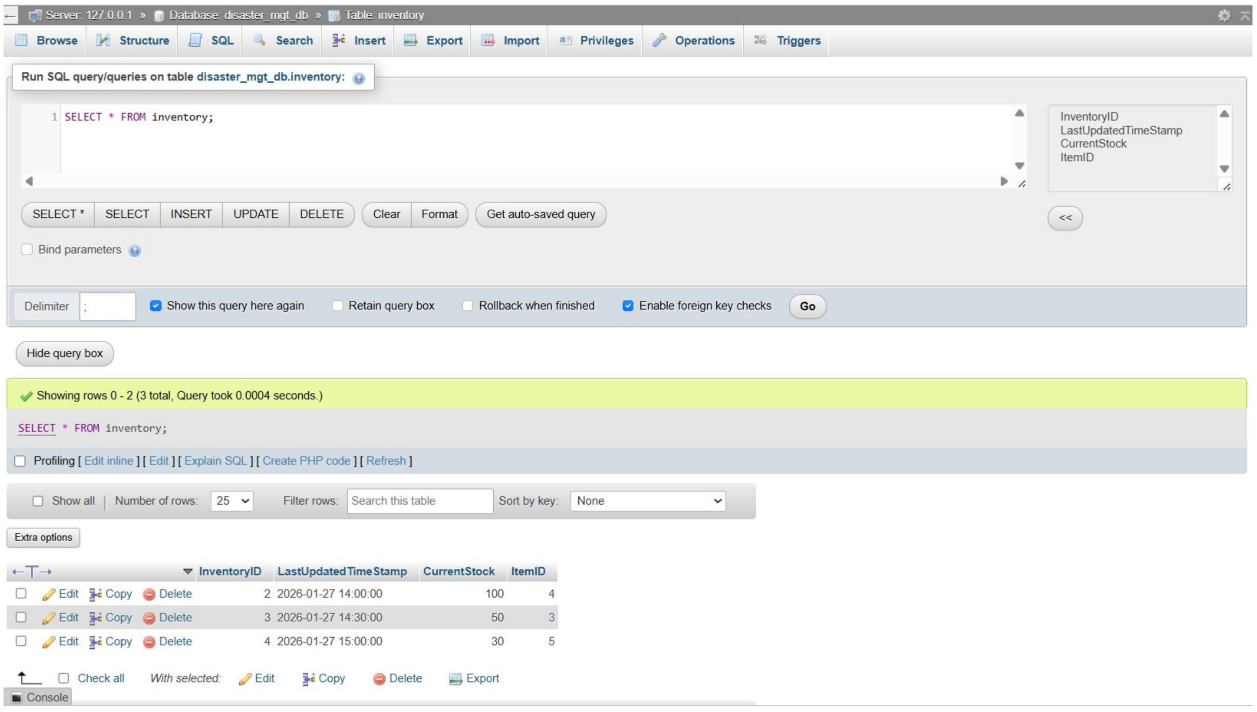
```
SELECT * FROM item;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

	ItemID	ItemName	Unit	Category
Edit Copy Delete	2	cereal	5 packs	cereal
Edit Copy Delete	3	Rice	kg	Food
Edit Copy Delete	4	Water Bottle	liters	Drink
Edit Copy Delete	5	Blanket	piece	Clothing

Console Check all With selected: Edit Copy Delete Export

Table: Inventory


Server: 127.0.0.1 » Database: disaster_mgt_db » Table: inventory
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Run SQL query/queries on table disaster_mgt_db.inventory:

```
SELECT * FROM inventory;
```

InventoryID
LastUpdatedTimeStamp
CurrentStock
ItemID
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
Bind parameters
Delimiter ;
Show this query here again
Retain query box
Rollback when finished
Enable foreign key checks
Go
Hide query box

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

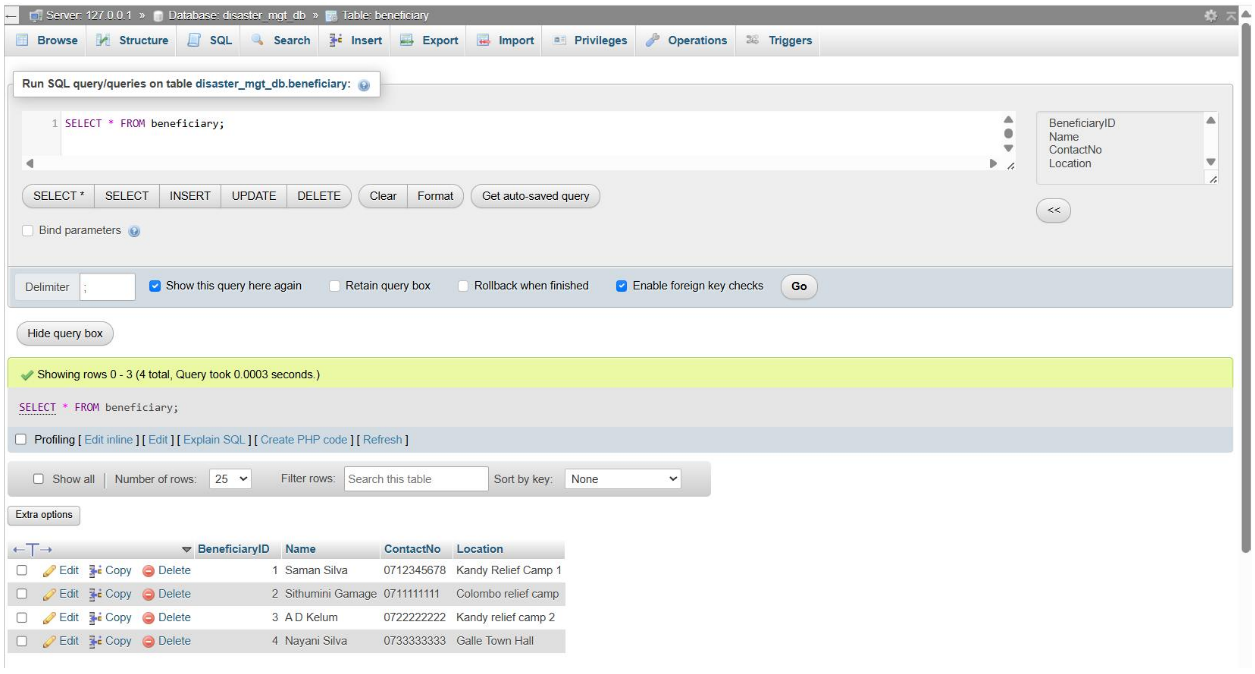
```
SELECT * FROM inventory;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

	InventoryID	LastUpdatedTimeStamp	CurrentStock	ItemID
Edit Copy Delete	2	2026-01-27 14:00:00	100	4
Edit Copy Delete	3	2026-01-27 14:30:00	50	3
Edit Copy Delete	4	2026-01-27 15:00:00	30	5

Check all With selected: Edit Copy Delete Export
Console

Table: Beneficiary


Server: 127.0.0.1 » Database: disaster_mgt_db » Table: beneficiary
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
Bind parameters
Hide query box
Extra options

Run SQL query/queries on table disaster_mgt_db.beneficiary:

```
SELECT * FROM beneficiary;
```

Columns: BeneficiaryID, Name, ContactNo, Location

Delimiter ; [✓ Show this query here again] Retain query box Rollback when finished [✓ Enable foreign key checks] Go

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

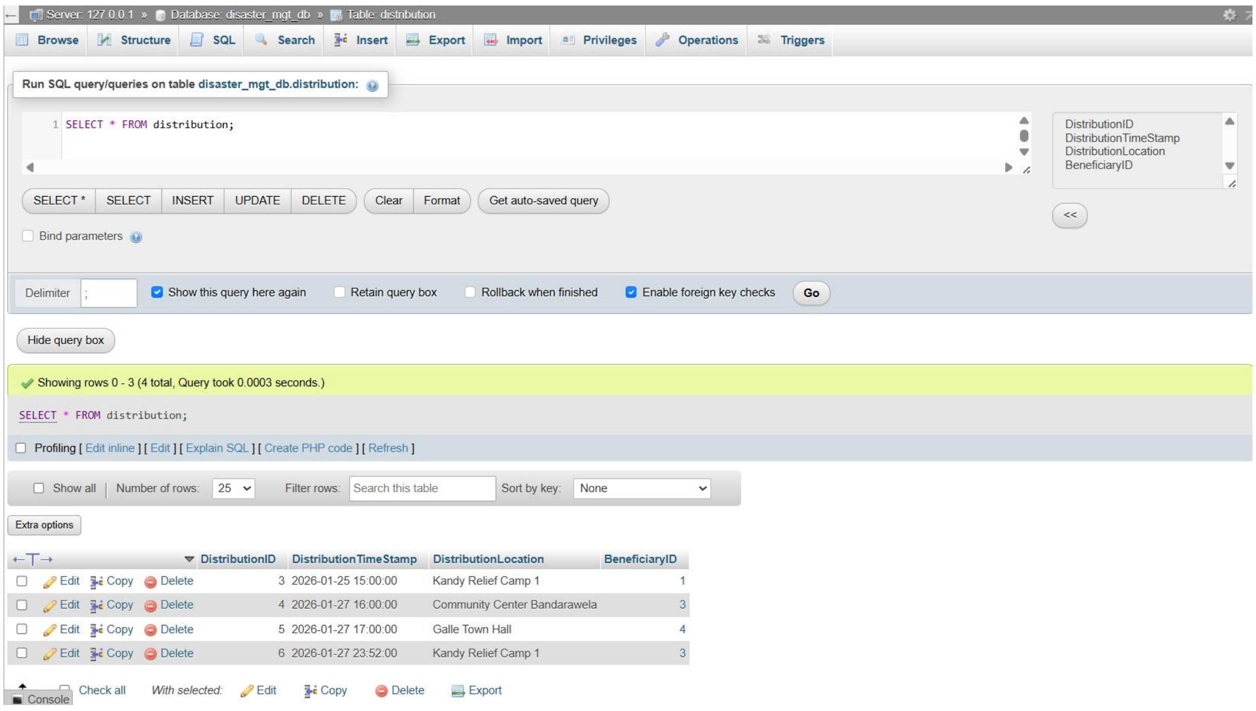
```
SELECT * FROM beneficiary;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

BeneficiaryID	Name	ContactNo	Location
1	Saman Silva	0712345678	Kandy Relief Camp 1
2	Sithumini Gamage	0711111111	Colombo relief camp
3	A D Kelum	0722222222	Kandy relief camp 2
4	Nayani Silva	0733333333	Galle Town Hall

Table: Distribution


Server: 127.0.0.1 » Database: disaster_mgt_db » Table: distribution
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
Bind parameters
Hide query box
Extra options
Check all With selected: Edit Copy Delete Export
Console

Run SQL query/queries on table disaster_mgt_db.distribution:

```
SELECT * FROM distribution;
```

Columns: DistributionID, DistributionTimeStamp, DistributionLocation, BeneficiaryID

Delimiter ; [✓ Show this query here again] Retain query box Rollback when finished [✓ Enable foreign key checks] Go

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

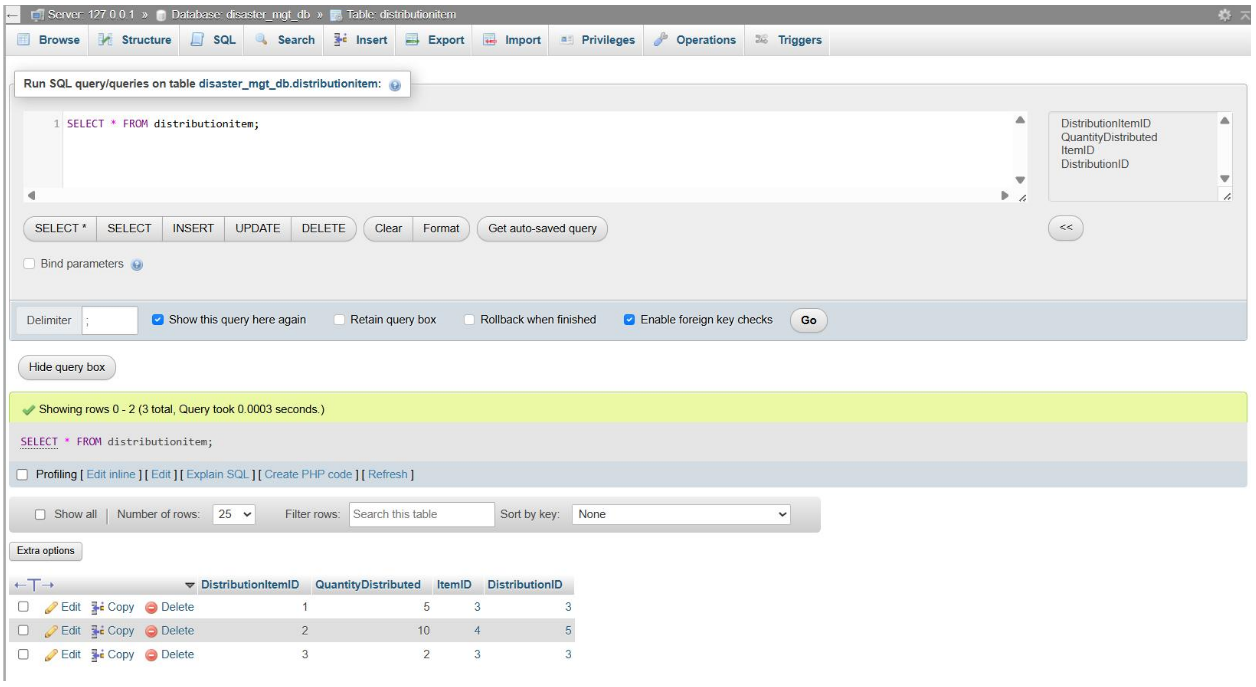
```
SELECT * FROM distribution;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

DistributionID	DistributionTimeStamp	DistributionLocation	BeneficiaryID
3	2026-01-25 15:00:00	Kandy Relief Camp 1	1
4	2026-01-27 16:00:00	Community Center Bandarawela	3
5	2026-01-27 17:00:00	Galle Town Hall	4
6	2026-01-27 23:52:00	Kandy Relief Camp 1	3

Table: DistributionItem


Server: 127.0.0.1 » Database: disaster_mgt_db » Table: distributionitem
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
Bind parameters
Hide query box
Show all Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

Run SQL query/queries on table disaster_mgt_db.distributionitem:

```
SELECT * FROM distributionitem;
```

DistributionItemID
QuantityDistributed
ItemID
DistributionID

Delimiter ; [✓] Show this query here again [ ] Retain query box [ ] Rollback when finished [✓] Enable foreign key checks Go

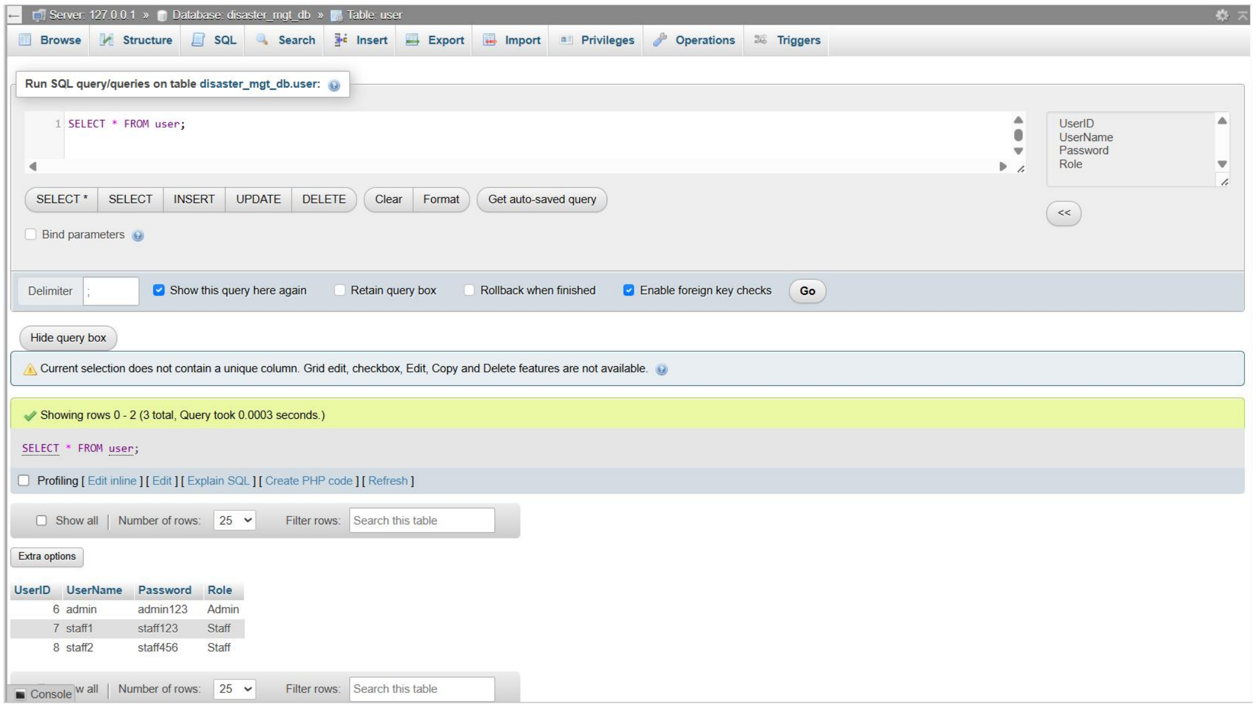
Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

```
SELECT * FROM distributionitem;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

	DistributionItemID	QuantityDistributed	ItemID	DistributionID
Edit Copy Delete	1	5	3	3
Edit Copy Delete	2	10	4	5
Edit Copy Delete	3	2	3	3

Table: User


Server: 127.0.0.1 » Database: disaster_mgt_db » Table: user
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
Bind parameters
Hide query box
Show all Number of rows: 25 Filter rows: Search this table
Extra options
Console

Run SQL query/queries on table disaster_mgt_db.user:

```
SELECT * FROM user;
```

UserID
UserName
Password
Role

Delimiter ; [✓] Show this query here again [ ] Retain query box [ ] Rollback when finished [✓] Enable foreign key checks Go

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

```
SELECT * FROM user;
```

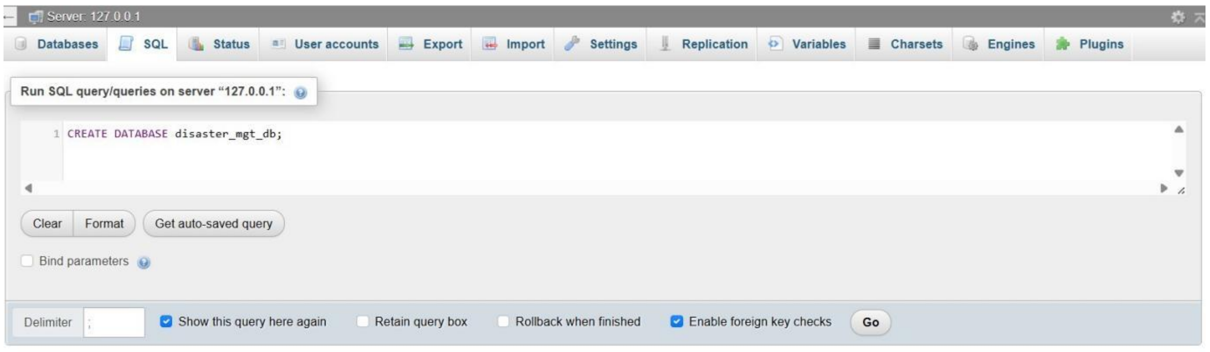
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

UserID	UserName	Password	Role
6	admin	admin123	Admin
7	staff1	staff123	Staff
8	staff2	staff456	Staff

7. Sample CRUD Operations

The following screenshots demonstrate the functional implementation of Create, Read, and Delete operations using SQL.

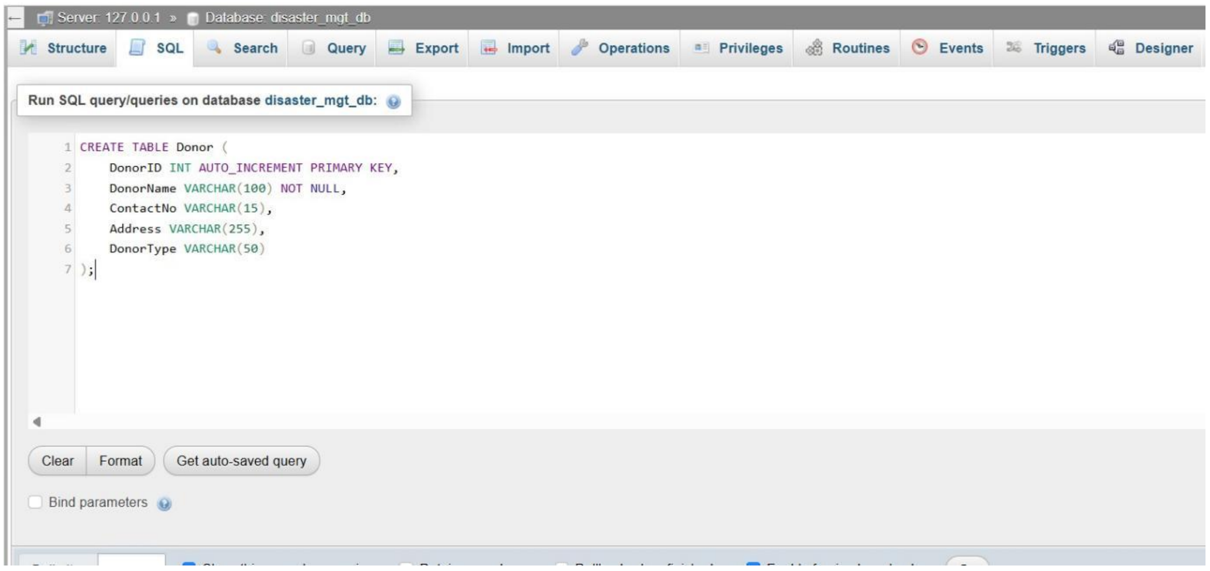
Database Creation



```
CREATE DATABASE disaster_mgt_db;
```

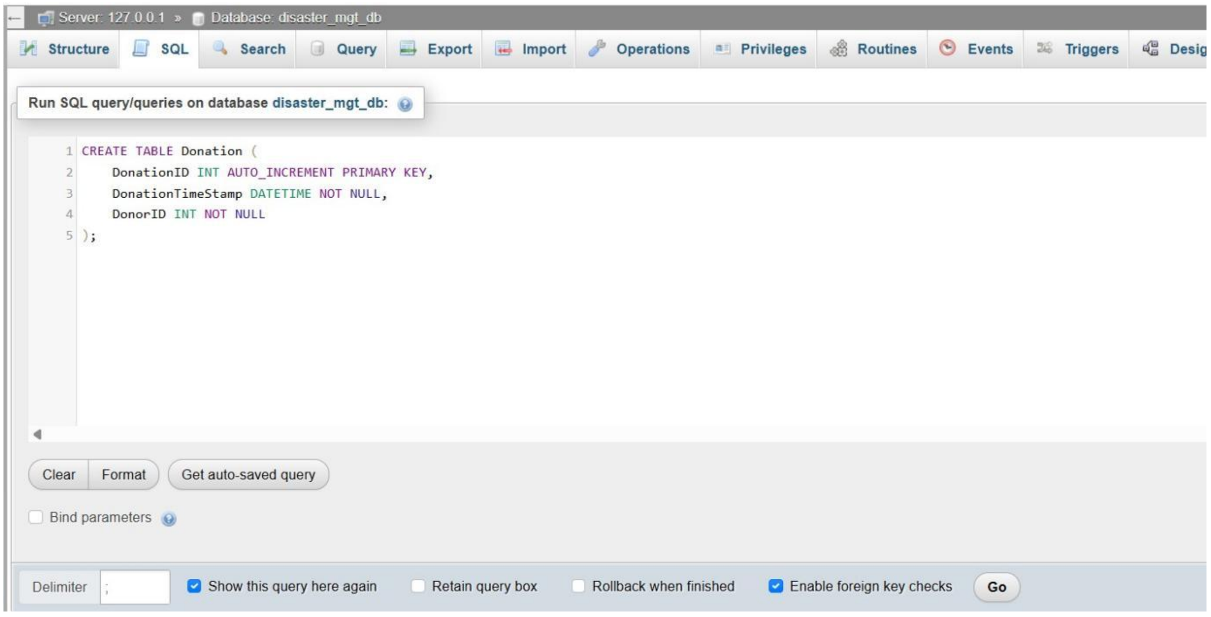
Table Creation

Table: Donor



```
CREATE TABLE Donor (
    DonorID INT AUTO_INCREMENT PRIMARY KEY,
    DonorName VARCHAR(100) NOT NULL,
    ContactNo VARCHAR(15),
    Address VARCHAR(255),
    DonorType VARCHAR(50)
);
```

Table: Donation


Server: 127.0.0.1 » Database: disaster_mgt_db
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Desig
Clear Format Get auto-saved query
Go

Run SQL query/queries on database disaster_mgt_db:

```sql
CREATE TABLE Donation (
    DonationID INT AUTO_INCREMENT PRIMARY KEY,
    DonationTimeStamp DATETIME NOT NULL,
    DonorID INT NOT NULL
);
```

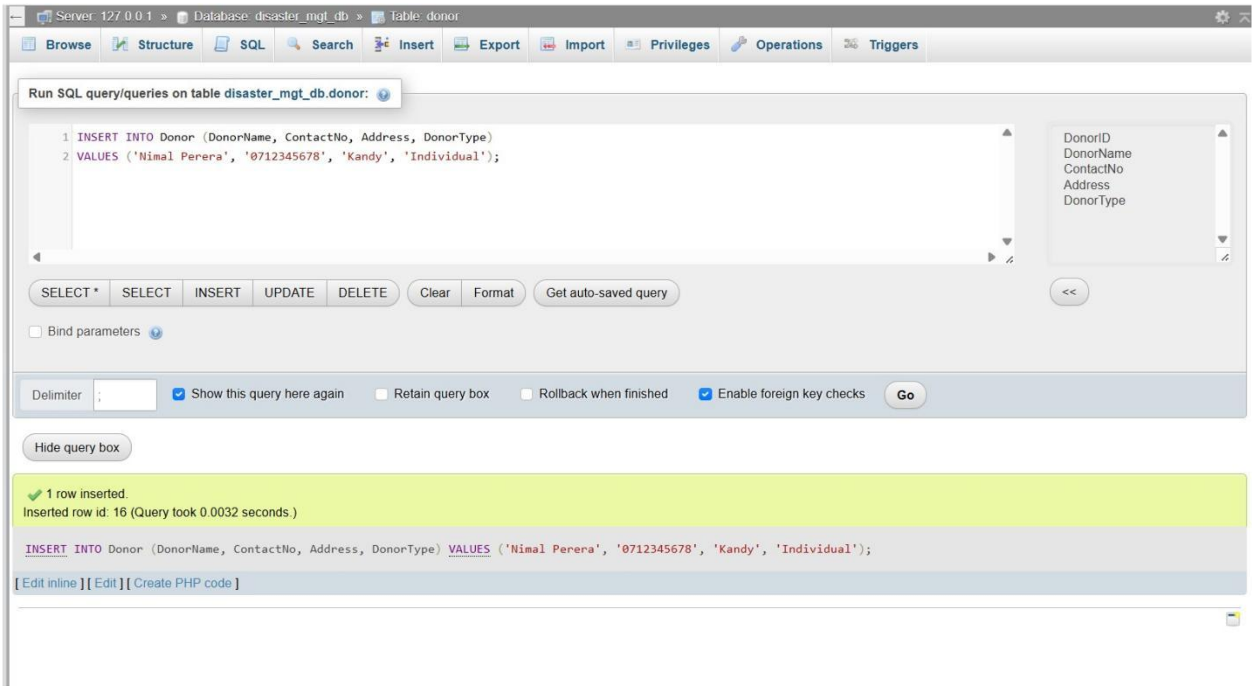
Bind parameters

Delimiter ; [✓ Show this query here again] Retain query box Rollback when finished [✓ Enable foreign key checks]

INSERT Operation (Create)

SQL Query used to insert a new donation record into the system.

Table: Donor


Server: 127.0.0.1 » Database: disaster_mgt_db » Table: donor
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
<<
Go
Hide query box
[Edit inline] [Edit] [Create PHP code]

Run SQL query/queries on table disaster_mgt_db.donor:

```sql
INSERT INTO Donor (DonorName, ContactNo, Address, DonorType)
VALUES ('Nimal Perera', '0712345678', 'Kandy', 'Individual');
```

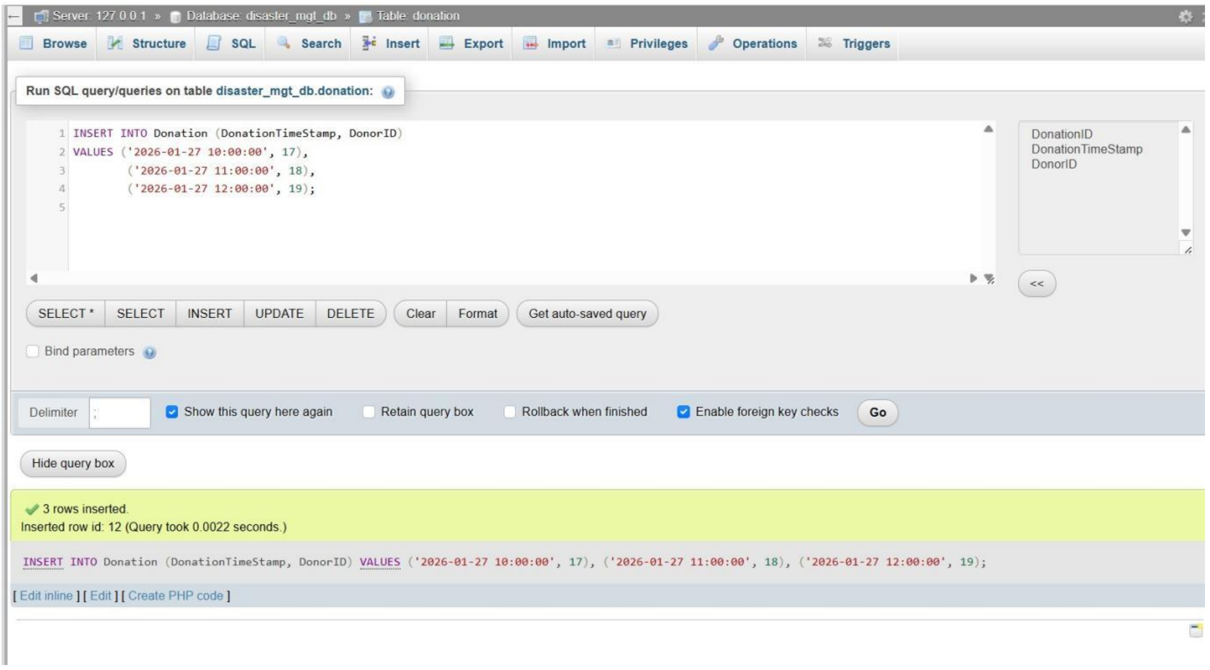
DonorID
DonorName
ContactNo
Address
DonorType

Bind parameters

Delimiter ; [✓ Show this query here again] Retain query box Rollback when finished [✓ Enable foreign key checks]

```
1 row inserted.
Inserted row id: 16 (Query took 0.0032 seconds.)
INSERT INTO Donor (DonorName, ContactNo, Address, DonorType) VALUES ('Nimal Perera', '0712345678', 'Kandy', 'Individual');
```

Table: Donation


Server: 127.0.0.1 » Database: disaster_mgt_db » Table: donation
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Run SQL query/queries on table disaster_mgt_db.donation:

```
INSERT INTO Donation (DonationTimeStamp, DonorID)
VALUES ('2026-01-27 10:00:00', 17),
       ('2026-01-27 11:00:00', 18),
       ('2026-01-27 12:00:00', 19);
```

DonationID
DonationTimeStamp
DonorID
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
Bind parameters
Delimiter ; Show this query here again Retain query box Rollback when finished Enable foreign key checks Go
Hide query box

3 rows inserted.
Inserted row id: 12 (Query took 0.0022 seconds.)

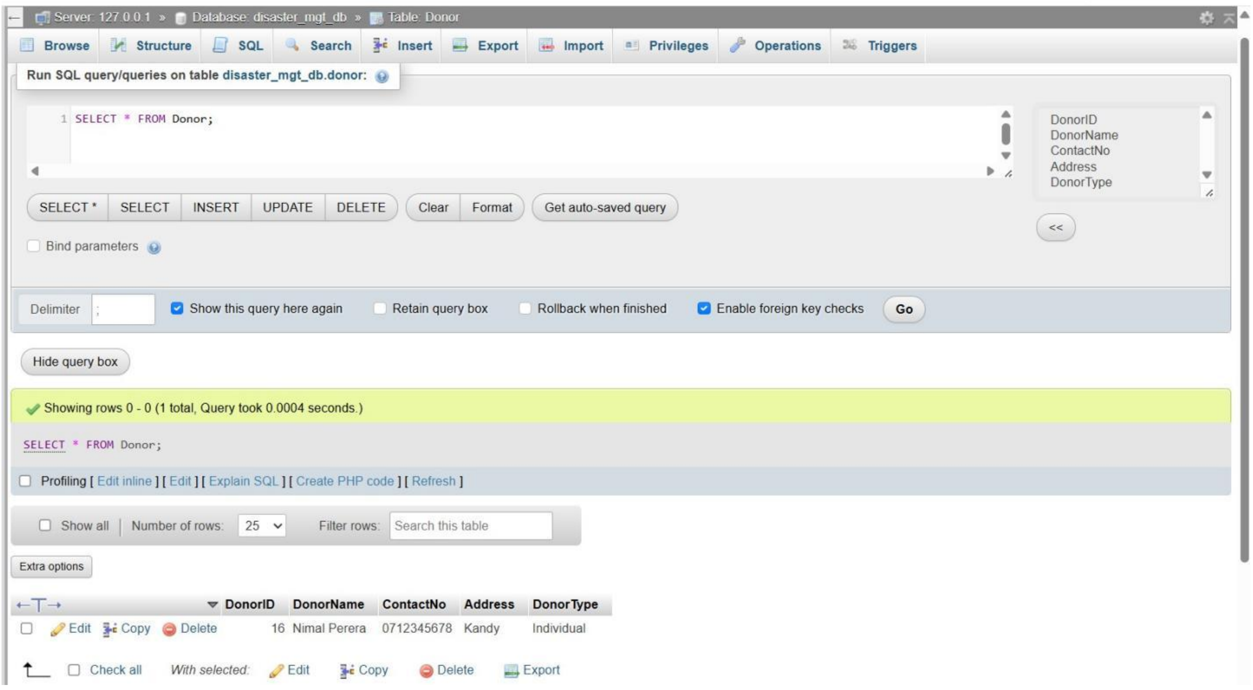
```
INSERT INTO Donation (DonationTimeStamp, DonorID) VALUES ('2026-01-27 10:00:00', 17), ('2026-01-27 11:00:00', 18), ('2026-01-27 12:00:00', 19);
```

[Edit inline] [Edit] [Create PHP code]

SELECT Operation (Read)

A complex query used to view specific data

Table: Donor


Server: 127.0.0.1 » Database: disaster_mgt_db » Table: Donor
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Run SQL query/queries on table disaster_mgt_db.donor:

```
SELECT * FROM Donor;
```

DonorID
DonorName
ContactNo
Address
DonorType
SELECT * SELECT INSERT UPDATE DELETE Clear Format Get auto-saved query
Bind parameters
Delimiter ; Show this query here again Retain query box Rollback when finished Enable foreign key checks Go
Hide query box

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

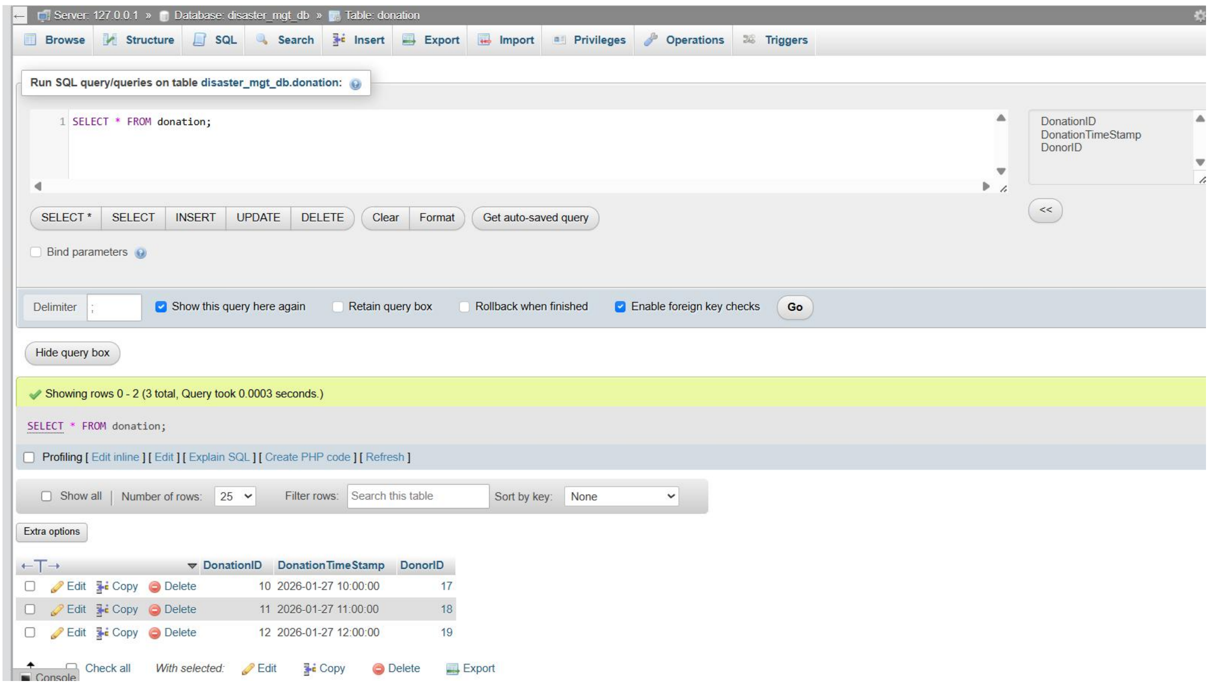
```
SELECT * FROM Donor;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table
Extra options

	DonorID	DonorName	ContactNo	Address	DonorType
Edit Copy Delete	16	Nimal Perera	0712345678	Kandy	Individual

Check all With selected: Edit Copy Delete Export

Table: Donation



Run SQL query/queries on table disaster_mgt_db.donation:

```
SELECT * FROM donation;
```

Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

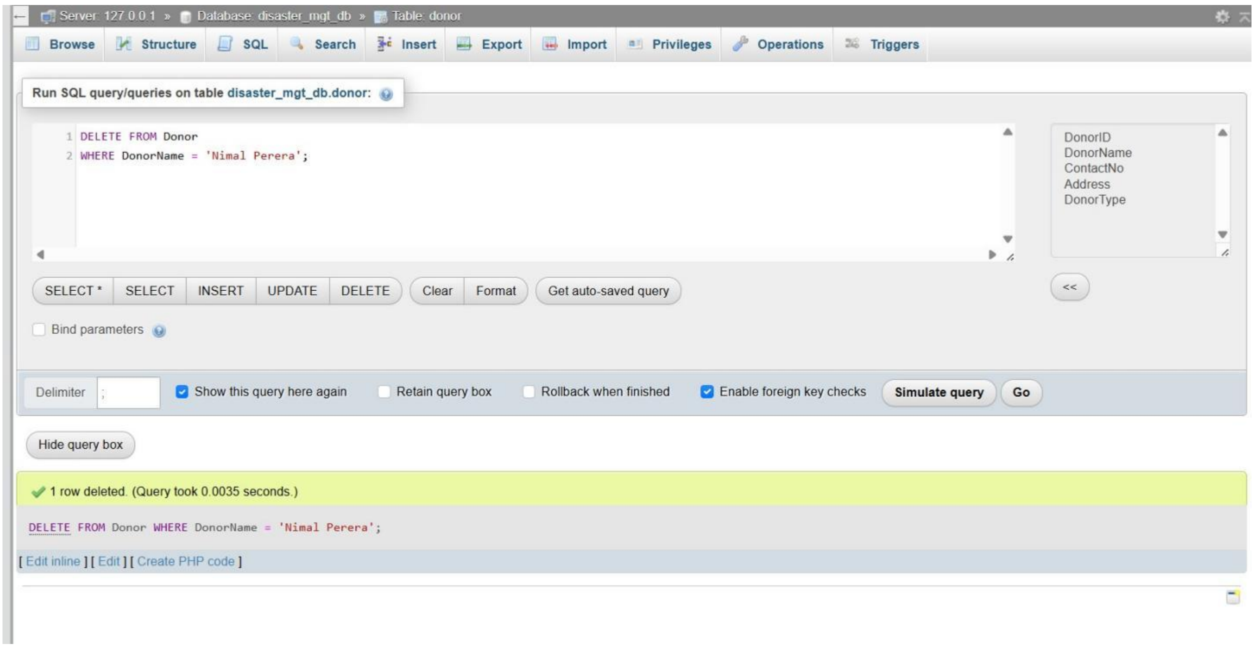
```
SELECT * FROM donation;
```

DonationID	DonationTimeStamp	DonorID
10	2026-01-27 10:00:00	17
11	2026-01-27 11:00:00	18
12	2026-01-27 12:00:00	19

DELETE Operation (Delete)

SQL Query demonstrating the removal of a record.

Table: Donor



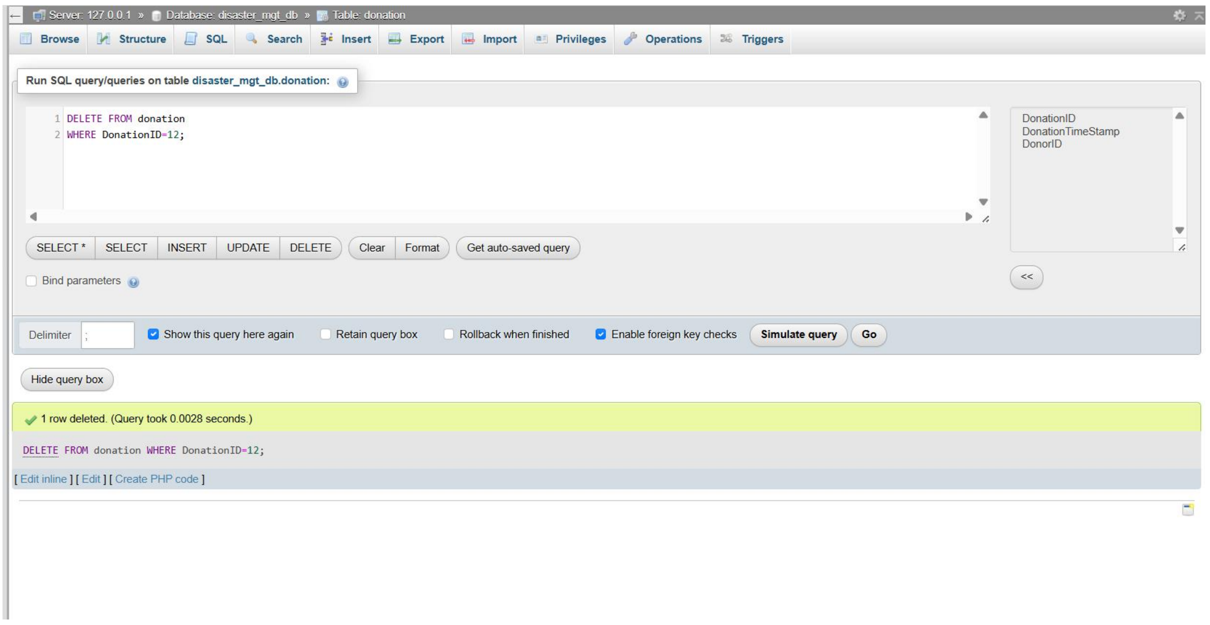
Run SQL query/queries on table disaster_mgt_db.donor:

```
DELETE FROM Donor
WHERE DonorName = 'Nimal Perera';
```

1 row deleted. (Query took 0.0035 seconds.)

```
DELETE FROM Donor WHERE DonorName = 'Nimal Perera';
```

Table: Donation



```
DELETE FROM donation
WHERE DonationID=12;
```

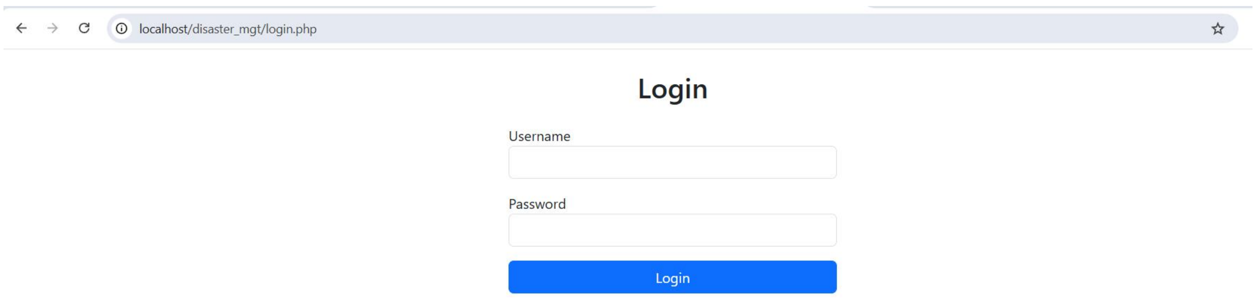
1 row deleted. (Query took 0.0028 seconds.)

```
DELETE FROM donation WHERE DonationID=12;
```

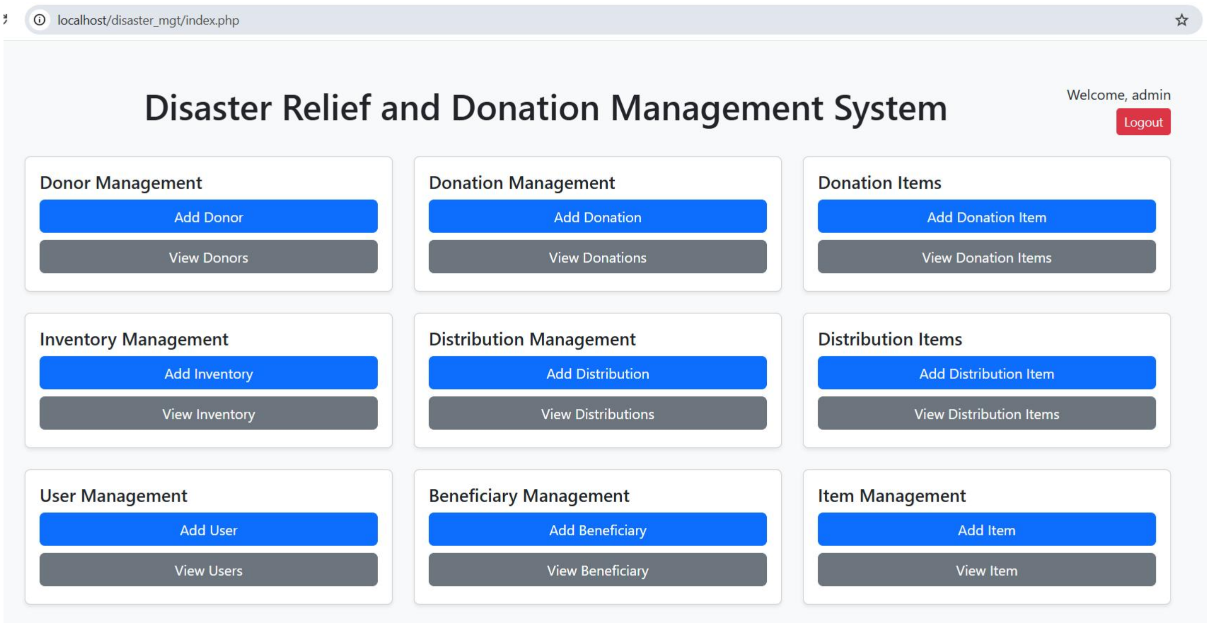
8. User Interface Implementation

The following screenshots illustrate the developed frontend application, which provides a user-friendly interface for the database.

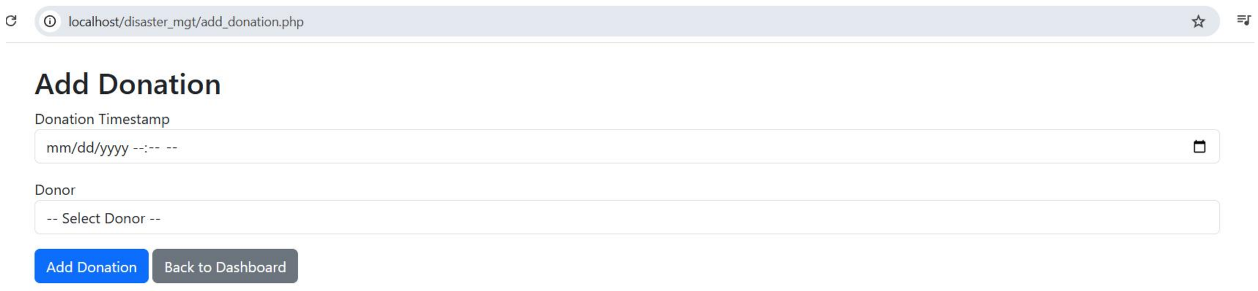
8.1 Login Interface



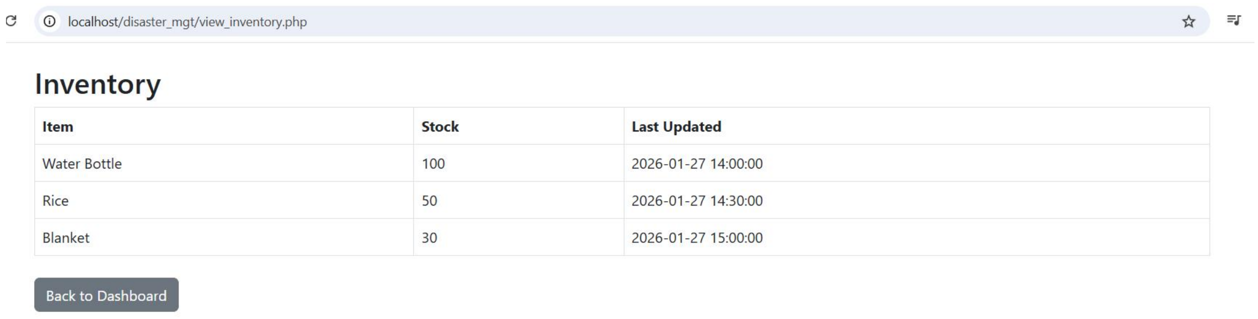
8.2 Dashboard



8.3 Donation Entry Form



8.4 Inventory View



Inventory

Item	Stock	Last Updated
Water Bottle	100	2026-01-27 14:00:00
Rice	50	2026-01-27 14:30:00
Blanket	30	2026-01-27 15:00:00

9. Conclusion

The **Disaster Relief and Donation Management System** has been successfully implemented, meeting the requirements of Phase 1, 2 and 3. By converting the conceptual ER model into a relational schema and implementing it within a DBMS, we have created a robust solution for tracking relief efforts. The system effectively handles data integrity through Foreign Key constraints, preventing orphaned records in critical areas like Donations and Distributions. The User Interface further enhances usability, allowing non-technical staff to perform CRUD operations without interacting directly with SQL code. This project demonstrates a complete lifecycle of database development, from problem definition to physical deployment.

10. Individual Contributions

The following table details the specific responsibilities and contributions of each group member towards the completion of the project.

Student ID	Name	Contribution	Signature
AS20240395	K I R Abeysekara	• Proposed the project topic and prepared the project proposal • Converted ER diagram into relational schema • Designed and implemented the complete database using phpMyAdmin • Implemented PHP-based UI • Provided necessary screenshots for the report. • Compiled and finalized the complete project report.	
AS20240427	B P R Sachintha	• Finalized the Project Proposal. • Designed the complete Entity-Relationship (ER) Diagram. • Identified entities, attributes, relationships, and cardinalities. • Conducted user interface research and provided feedback. • Assisted in preparing and reviewing SQL queries • Performed System Testing and Data Validation.	
AS20240404	A H M D Silva	• Assisted in analyzing system requirements and project scope • Assisted in converting the ER diagram into relational tables • Reviewed primary key and foreign key mappings • Performed System Testing and Data Validation. • Assisted in reviewing the final report.	
AS20240577	N G S Sathsara	• Reviewed screenshots and formatting consistency • Assisted in database testing	

AS20240546	T H Sandaruwan	• Assisted in structuring the project report • Assisted in database testing	Sandaruwan
AS20240559	E K S Govinda	• Performed System Testing and Data Validation. • Assisted in final documentation review	
AS20240538	S M Wijesekara	• Adding table of contents and formatting for the final report.	[illegible]

Group Members

AS20240395-K I R Abeysekara

AS20240427- B P R Sachintha

AS20240404- A H M D Silva

AS20240577- N G S Sathsara

AS20240546- T H Sandaruwan

AS20240559- E K S Govinda

AS20240538- S M Wijesekara