Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Semester: (Fall, Year: 2024), B.Sc. in CSE (Day)

Disaster Relief Management System

Course Title: Database System Lab

Course Code: CSE 210

Section: 231 D1

<u>Students Details</u>

Name	ID
Asfiqul Alam Emran	231902049

Submission Date: 15/12/2024

Course Teacher's Name: Fatema Tuj Johora

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	Signature:
Comments:	Date:

Contents

Chapter 1

Introduction

1.1 Overview

The Disaster Relief Management System is designed to streamline and improve the coordination of disaster relief efforts. This system provides a centralized platform to manage and track relief resources, organize distribution schedules, and monitor beneficiaries' needs. By utilizing a robust database, the system enables relief agencies, donors, and volunteers to coordinate their activities efficiently, ensuring that supplies and aid reach the right people at the right time.

1.2 Motivation

This project was motivated by the need for a streamlined approach to manage disaster relief resources and enhance communication among relief agencies, donors, and volunteers. By developing a centralized database, this system can help ensure that aid reaches those in need promptly and accurately. The goal is to create a reliable platform that supports relief efforts, minimizes logistical issues, and ultimately aids in reducing the impact of disasters on affected communities.

1.3 Problem Definition

1.3.1 Problem Statement

The Disaster Relief Management System aims to solve these problems by creating a central database that helps relief organizations track resources, plan distributions, and quickly respond to those in need. This system is designed to make disaster relief efforts faster, more efficient, and better organized, helping affected communities recover more effectively.

1.3.2 Complex Engineering Problem

The coordination of disaster relief efforts involves handling a large volume of dynamic data, including information about relief agencies, supplies, beneficiaries, locations, and distribution schedules. The challenge lies in designing a system that can manage these various data sets in real-time, ensure accuracy in resource allocation, and maintain scalability as the scope of a disaster evolves. One of the key engineering problems in this project is the integration of diverse data sources, while maintaining data consistency and integrity across multiple entities involved in the relief process. Additionally, the system must handle concurrent updates, support rapid data retrieval, and offer a robust mechanism for conflict resolution when different agencies input overlapping or contradictory information. This requires an optimized database structure that can support complex relationships, handle large-scale queries efficiently, and ensure the system remains responsive under heavy load during disaster scenarios.

The solution must also address data security and access control, ensuring that sensitive information is protected while allowing authorized personnel to collaborate seamlessly. Developing a system that balances performance, reliability, and security under real-world conditions presents a significant engineering challenge in this disaster relief management project.

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributess	Explain how to address
P1: Depth of knowledge required	Address by providing comprehensive documentation, tutorials, and training materials to users, developers, and stakeholders.
P2: Range of conflicting requirements	Address by conducting thorough requirement analysis and stakeholder consultations to identify conflicting requirements early in the development process.

P3: Depth of analysis required	Address by breaking down complex problems into smaller, more manageable components and conducting systematic analysis at each level.
P4: Familiarity of issues	Address by providing training, workshops, and resources to familiarize team members with relevant concepts, technologies, and industry practices.
P5: Extent of applicable codes	Address by conducting a thorough review of applicable codes, standards, regulations, and industry guidelines relevant to the project.
P6: Extent of stakeholder involvement and conflicting requirements	Address by establishing effective communication channels and collaboration framework to engage stakeholders throughout the project life cycle.
P7: Interdependence	Address by identifying and mapping interdependence between project components, tasks, and stakeholders.

1.4 Design Goals/Objectives

1. Efficient Data Management: Create a centralized database structure that can store, manage, and retrieve large volumes of data related to relief agencies, supplies, beneficiaries, and distribution schedules. The system should facilitate quick and accurate data access to aid in decision-making during disaster response.

2. Scalability: Design the system to handle increased data loads and user demands as the scale and severity of a disaster grows. The database should support expansion, allowing new tables, fields, or data types to be added as needed without major structural changes.

3. Real-Time Tracking and Updates: Enable real-time tracking of resources and immediate updates to the database to reflect current inventory levels, distribution plans, and beneficiary needs. This ensures that the most accurate information is available to relief coordinators at all times.

4. Data Integrity and Consistency: Implement mechanisms to maintain data accuracy, avoid duplication, and ensure consistency across multiple data entries. The system should support concurrent data entry and updates without compromising the integrityof the data.

5. User Access Control and Security: Establish strict access control measures to protect sensitive information while allowing authorized personnel to perform their tasks. Data security protocols will ensure that only trusted users can access, modify, or manage critical information.

6. Resource Allocation Optimization: Design the system to assist in the optimal allocation and distribution of relief resources, helping minimize waste and ensure resources reach the areas of greatest need. The system should facilitate planning and tracking to enhance the effectiveness of disaster response efforts.

1.5 Application

The Railway Database Management System (RDMS) is a vital application in modern railway operations, enhancing the efficiency, safety, and convenience of train services. It is designed to manage vast amounts of data related to train schedules, passenger bookings, freight services, and station operations. In the real world, RDMS ensures accurate tracking of train movements, helping to prevent collisions and delays through real-time data sharing between stations and trains. It also streamlines the ticketing process, al lowing passengers to book tickets, check seat availability, and receive real-time updates about their journeys. Additionally, it facilitates effective resource allocation for railway personnel and maintenance, contributing to overall system optimization. Through centralized data management, RDMS improves the coordination between various railway departments, enhancing operational efficiency and customer satisfaction.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

Start the section with a general discussion of the project [1] [2] [3].

2.2 Project Details

This project involves designing and implementing a MySQL database system for disaster and resource management. The database consists of multiple tables, including event_logs, logistics, disasters, and donations, each serving a distinct purpose in tracking disaster-related events, managing resources, and recording donations.

The system demonstrates various database functionalities, such as table creation, data population, and SQL operations like filtering, logical conditions, joins, and aggregate functions. Advanced features like triggers were also implemented to automate logging activities, ensuring consistency and reducing manual intervention.

The project simulates real-world disaster management scenarios, emphasizing effective data organization and retrieval to support decision-making and efficient resource allocation.

2.3 Implementation

```
CREATE TABLE disasters (
   id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    name VARCHAR(255),
    type VARCHAR(255),
    date DATE,
    location VARCHAR(255),
    severity VARCHAR(50)
);

CREATE TABLE donations (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    amount DECIMAL(10, 2),
    date DATE,
    FOREIGN KEY (user_id) REFERENCES users(id)
);

CREATE TABLE event_logs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    event_type VARCHAR(255),
    description TEXT,
    date DATE
);

CREATE TABLE logistics (
    id INT AUTO_INCREMENT PRIMARY KEY,
    resource_id INT,
    quantity INT,
    status VARCHAR(50),
    date DATE,
    FOREIGN KEY (resource_id) REFERENCES resources(id)
);

CREATE TABLE messages (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    message TEXT,
    date DATE,
```

```
    FOREIGN KEY (user_id) REFERENCES users(id)
);

CREATE TABLE relief_terms (
    id INT AUTO_INCREMENT PRIMARY KEY,
    term VARCHAR(255),
    definition TEXT
);

CREATE TABLE resources (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    type VARCHAR(255),
    quantity INT
);

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    email VARCHAR(255),
    password VARCHAR(255)
);

CREATE TABLE volunteers (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    contact VARCHAR(255),
    availability VARCHAR(255)
);

CREATE TABLE volunteer_tasks (
    id INT AUTO_INCREMENT PRIMARY KEY,
    volunteer_id INT,
    task_description TEXT,
    status VARCHAR(50),
    date DATE,
```

```sql
    FOREIGN KEY (volunteer_id) REFERENCES volunteers(id)
);

INSERT INTO disasters (name, type, date, location, severity) VALUES
('Flood', 'Natural', '2023-01-15', 'City A', 'High'),
('Earthquake', 'Natural', '2023-02-20', 'City B', 'Severe'),
('Fire', 'Accidental', '2023-03-12', 'City C', 'Moderate')

INSERT INTO users (name, email, password) VALUES
('Emran', 'emran@example.com', 'password1'),
('Talha', 'talha@example.com', 'password2'),
('Ruhin', 'ruhin@example.com', 'password3')

INSERT INTO donations (user_id, amount, date) VALUES
(1, 100.00, '2023-01-01'),
(2, 200.00, '2023-01-15'),
(3, 150.00, '2023-02-10')

INSERT INTO event_logs (event_type, description, date) VALUES
('Disaster Declared', 'A disaster has been declared in City A due to flooding.', '2023-01-
15'),
('Evacuation', 'Evacuation orders issued for City B after earthquake.', '2023-02-20'),
('Fire Contained', 'Fire in City C has been contained.', '2023-03-12')

INSERT INTO logistics (resource_id, quantity, status, date) VALUES
(1, 50, 'Delivered', '2023-01-02'),
(2, 100, 'In Transit', '2023-01-16'),
(3, 70, 'Delivered', '2023-02-11')

INSERT INTO messages (user_id, message, date) VALUES
(1, 'We need more volunteers.', '2023-01-01'),
(2, 'Donation received, thank you!', '2023-01-15'),
(3, 'Relief materials dispatched.', '2023-02-10')

INSERT INTO relief_terms (term, definition) VALUES
('Evacuation', 'The process of moving people from a dangerous place to safety.'),
('Shelter', 'A place providing temporary protection from bad weather or danger.'),
('Rescue', 'To save someone from a dangerous or distressing situation.')


INSERT INTO resources (name, type, quantity) VALUES
('Water Bottles', 'Consumable', 500),
('Blankets', 'Non-Consumable', 200),
```

('Medical Kits', 'Medical', 150)

INSERT INTO volunteers (name, contact, availability) VALUES
('Shanto', 'shanto@example.com', 'Weekends'),
('Fardin', 'fardin@example.com', 'Weekdays'),
('Tanvir', 'tanvir@example.com', 'Weekends')

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

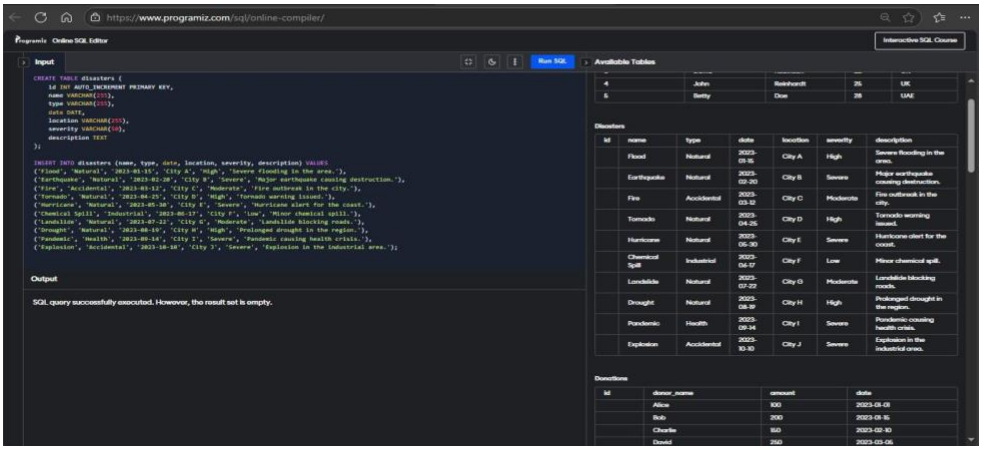

Figure 3.1: login query

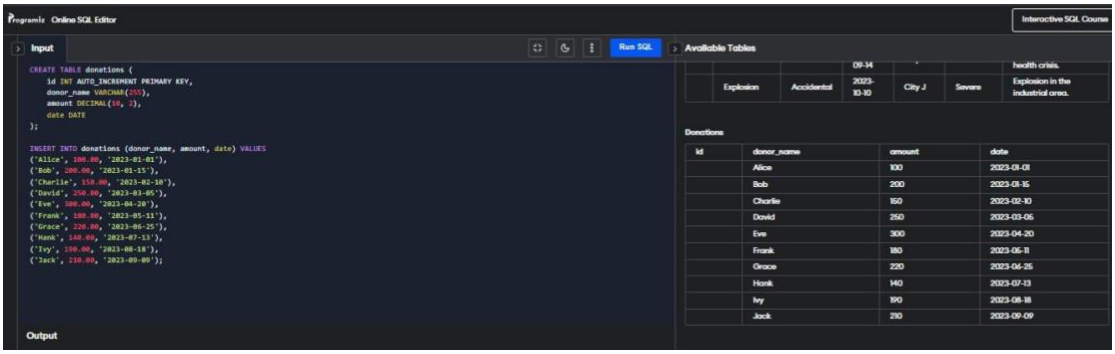

Figure 3.2: user details query

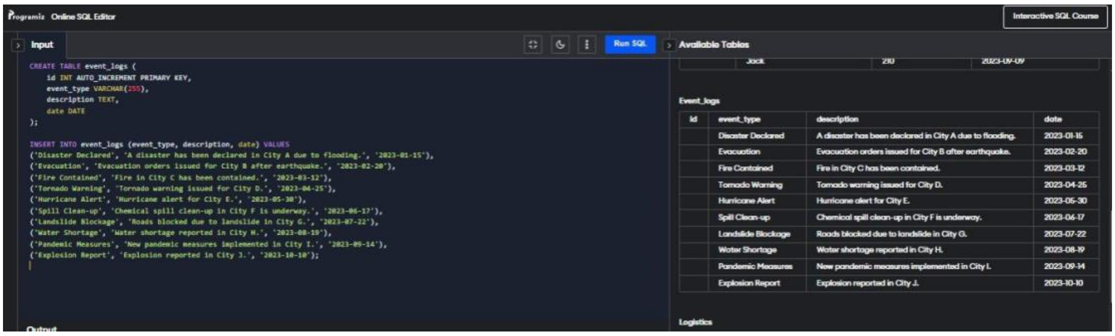

Figure 3.3 Event login

3.2 Results Analysis/Testing

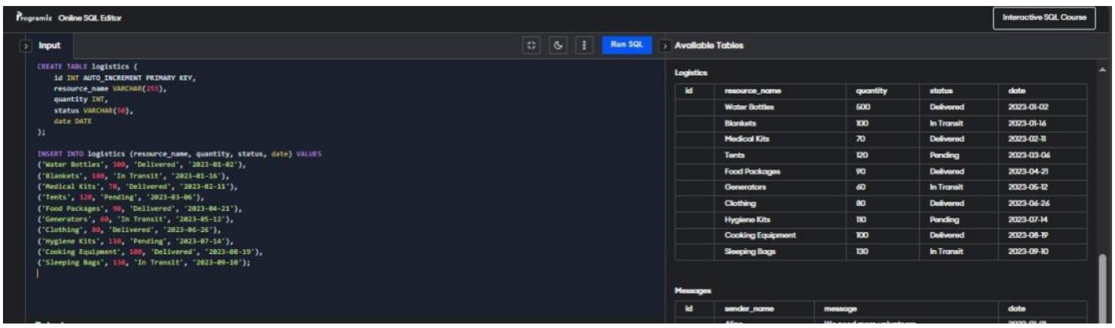

Figure : logistics

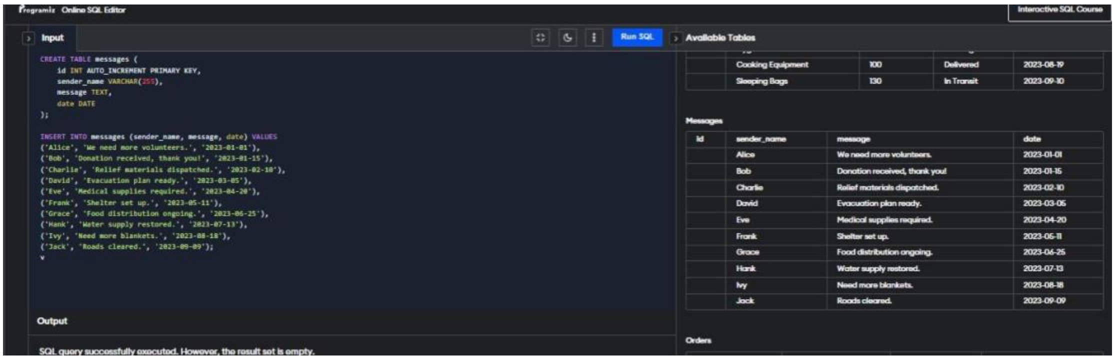

Figure : Message

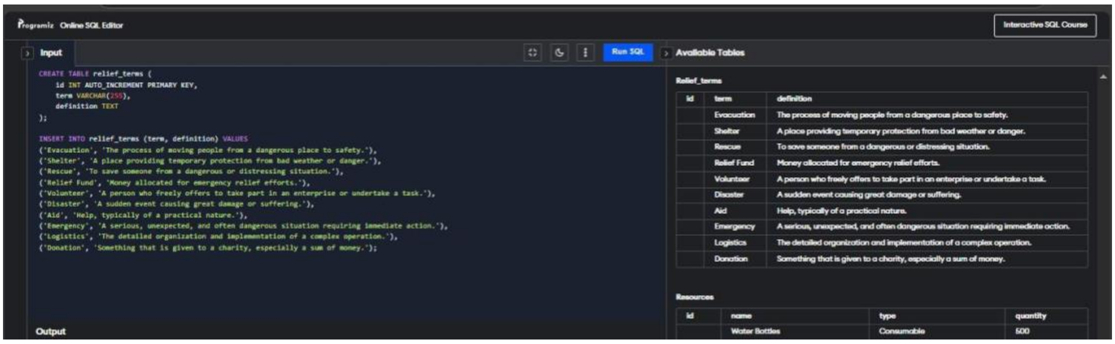

Figure : reilf teams

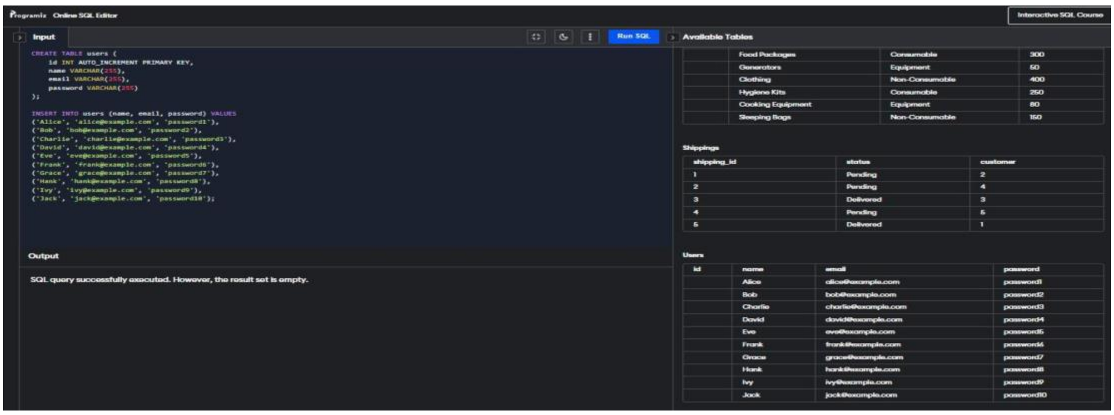

Figure : Users

3.3 Results Overall Discussion

This project involved designing and implementing a MySQL database system comprising multiple tables (event_logs, logistics, etc.) and performing various database operations. The objective was to demonstrate practical applications of database management concepts, including schema creation, data manipulation, and advanced SQL functionalities like triggers, joins, and aggregate functions.

Chapter 4

Conclusion

4.1 Discussion

This project successfully demonstrated the creation and manipulation of a MySQL database system, showcasing schema design, data operations, and advanced features like triggers and joins. The integration of logical operations and aggregate functions provided valuable insights into resource management and event tracking. The database effectively simulates real-world disaster management scenarios.

4.2 Limitations

The project avoided foreign key relationships, limiting relational integrity between tables. Additionally, the dataset was manually generated, which may not reflect real-world data complexity. Triggers, while useful, could lead to maintenance challenges in larger systems.

4.3 Scope of Future Work

Future improvements could include integrating foreign keys for better relational consistency, expanding datasets with real-world inputs, and implementing a user interface for better accessibility. Incorporating machine learning could further enhance decision-making in disaster management.

References

[1] Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, and Vishanth Weerakkody. Critical analysis of big data challenges and analytical methods. *Journal of Business Research*, 70:263–286, 2017.

[2] Douglas Laney. 3d data management: controlling data volume, velocity and variety. gartner, 2001.

[3] MSWindows NT kernel description. http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm. Accessed Date: 2010-09-30.