**TT HOLDING DATABASE DESIGN REPORT**

**Table of Contents**

[Figures Index 4](#_Toc194039312)

[Tables Overview 4](#_Toc194039313)

[Abbreviations Used 5](#_Toc194039314)

[Abstract 5](#_Toc194039315)

[Chapter 1: Introduction 6](#_Toc194039316)

[1.2 Problem Solving 6](#_Toc194039318)

[1.3 Objective 6](#_Toc194039319)

[1.4 Scope & Constraints 6](#_Toc194039320)

[Chapter 2: Literature Review. 7](#_Toc194039321)

[2.1 Introduction of it 7](#_Toc194039322)

[2.2 Reviewing the Literature 7](#_Toc194039324)

[Review of relational database models for employment systems: 7](#_Toc194039325)

[2.3 Findings & Discussion 7](#_Toc194039326)

[Chapter 3: Methodology 8](#_Toc194039327)

[3.1 Requirement Analysis 8](#_Toc194039328)

[3.2 Design of the System 8](#_Toc194039329)

[3.2.2 UML Diagrams 8](#_Toc194039330)

[Entity relationship diagram (ERD) 8](#_Toc194039331)

[Multiple entities related to employment tracking: 9](#_Toc194039332)

[3.3 System Implementation / Prototyping 10](#_Toc194039333)

[MySQL database implementation 10](#_Toc194039334)

[Creation of Database and tables 11](#_Toc194039335)

[Data insertion 12](#_Toc194039336)

[Advanced SQL (Views, Triggers, Functions and 2 Procedures). 13](#_Toc194039337)

[Views creation 13](#_Toc194039338)

[Triggers Implementation 14](#_Toc194039339)

[3.4 Testing 15](#_Toc194039340)

[Test Triggers by Performing Relevant Actions 15](#_Toc194039341)

[Functions 16](#_Toc194039342)

[Procedures 16](#_Toc194039343)

[User Creation and Privilege Assignment 17](#_Toc194039344)

[Chapter 4: System Initiation and Planning 18](#_Toc194039345)

[4.1 Assessing Feasibility of the Project: 18](#_Toc194039346)

[4.2 Project Plan 19](#_Toc194039347)

[Chapter 5: System Analysis 19](#_Toc194039348)

[5.1 Determining System Requirements 19](#_Toc194039349)

[Identifying Key Data Entities 19](#_Toc194039350)

[5.2 Structuring System Requirements 20](#_Toc194039351)

[Chapter 6: Conclusion 20](#_Toc194039352)

[6.1 The Advantages of the System 20](#_Toc194039353)

[6.2 The Future Enhancement of the System 20](#_Toc194039354)

[6.3 The Potential Benefits 20](#_Toc194039355)

[6.4 Conclusion 20](#_Toc194039356)

[References 22](#_Toc194039357)

[Appendices 23](#_Toc194039358)

# Figures Index

* *Figure 1.0 Crow’s foot ERD.*
* *Figure 1.1 Creation of database and first 4 tables.*
* *Figure 1.2 Creation of next 3 tables continued.*
* *Figure 1.3 insertion of 10 rows in first 3 tables.*
* *Figure 1.4 insertion of next 2 tables.*
* *Figure 1.5 insertion on last 2 tables.*
* *Figure 2.0 creation of 2 views.*
* *Figure 3.0 first 2 insert triggers.*
* *Figure 3.1 4 triggers (2 for update and 2 for delete).*
* *Figure 3.2 trying to insert a duplicate phone number and Preventing Salary Updates Below a Minimum Value.*
* *Figure 4.0 calling the function and get the total number of employees in the person table.*
* *Figure 4.1 function getting the total number of organizations in the organization table.*
* *Figure 5.0 The add\_person procedure and add\_organization procedure.*
* *Figure 5.1 creation of 2 users and granting them privileges.*

# Tables Overview

* + Person
  + Organization
  + Position
  + Position\_Held
  + Employment
  + Permanent\_Employee
  + Part\_Time\_Employee
  + organization\_view
  + position\_held\_view

# Abbreviations Used

* SQL – Structured Query Language
* ERD – Entity Relationship Diagram
* RDBMS – Relational Database Management System
* FK – Foreign Key
* PK – Primary Key

# Abstract

Large amounts of data in one place. With databases, organizations can quickly access, manage, modify, update, organize and retrieve their data (Alexander, 2024).

The TT Holding Database Design project’s aims to develop a database system that effectively manages employment relationships, positions, and organizations. The system ensures efficient tracking or retrieving of employees working across multiple organizations, whether as permanent or part-time employees. The report details the design, implementation, and testing of the system using MySQL, incorporating ER diagrams, and advanced SQL features such as triggers, stored procedures, and views. Additionally, it covers data insertion, validation, and testing to ensure integrity and performance.

# Chapter 1: Introduction

## 1.1 Problem Statement To handle staff members who may work for several companies and occupy various roles over time, organizations require an organized database. It is difficult to keep track of work roles, salary, and employment history when there is no effective system in place.

## 1.2 Problem Solving

This project introduces a relational database system that efficiently models employment, organizations, and job positions while ensuring data integrity through foreign key constraints.

## 1.3 Objective

* To design an ER/EER diagram mapping employment relationships.
* To create and implement MySQL database tables with relationships.
* Data insertion between 10 and 20 rows per table.
* To apply advanced SQL techniques such as views, triggers, functions, and procedures.
* To ensure user privileges and data security.

## 1.4 Scope & Constraints

* **Scope**: Covers employment tracking, organization relationships, and job positions.
* **Constraints**: Limited to MySQL implementation without external database integrations.

# Chapter 2: Literature Review.

## 2.1 Introduction of it

## Research for job tracking databases, relational database management systems (RDBMS), and database design. It focuses on the most effective methods for organizing employee and organizational data.

## 2.2 Reviewing the Literature

### Review of relational database models for employment systems:

Fundamentals of Database Systems

* This book explains ER modeling and database relational schema design, which were applied in designing the TT Holdings Database to avoid redundancy and maintain data integrity (Shamkani, 2020).

SQL Triggers and Stored Procedures in Database Security

* The research highlights the use of SQL triggers and stored procedures for data security and automation. This approach was applied by implementing triggers for insert, update, and delete operations, ensuring data consistency.

Foreign key Constraint

* Foreign key constraints are used to link the columns of two tables together, ensuring the referential integrity of the data. In essence, indicating that the rows in these two tables are related (Francois, 2024).
* Advanced SQL techniques in database optimization.

## 2.3 Findings & Discussion

* The literature suggests that relational databases effectively manage employment relationships.
* EER diagrams improve data representation for complex relationships.
* Advanced SQL features (triggers, procedures) enhance data integrity and automation.

# Chapter 3: Methodology

All SQL commands used for creating tables, triggers, functions, and procedures are documented in a separate notepad file, which is included in the appendices (Appendix 0). This file serves as a reference for the full list of SQL statements executed during the development of the TT Holding Database System.

## 3.1 Requirement Analysis

* Identification of system requirements.
* Understanding user needs for employment and organization tracking.
* Defining database schema and relationships.

3.2 Design of the System3.2.1 Design for Architecture

**Example of Tables and Relationships together with their cardinality:**

|  |  |  |
| --- | --- | --- |
| **ENTITIES** | **RELATIONSHIP** | **CARDINALITY** |
| Person – Employment | Employment | Many-to-Many |
| Employment –Organization | Have employees | Many-to-Many |
| Person – Position\_Held | Hold position | Many-to-Many |
| Organization – Position | Have positions | One-to-Many |
| Person – Permanent\_Employee | Be Permanent | One-to-One |
| Person – Part\_Time\_Employee | Be Part timed | One-to-One |

### 3.2.2 UML Diagrams

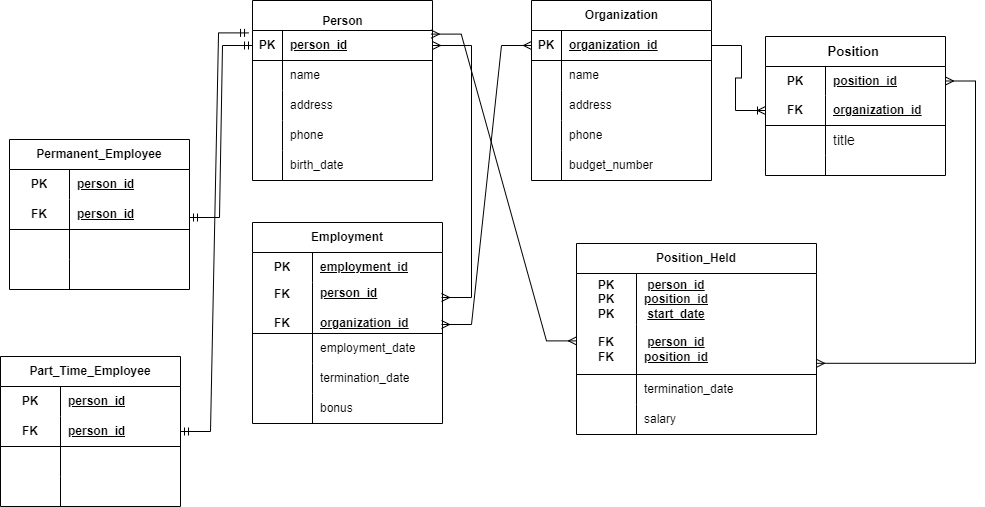
### Entity relationship diagram (ERD)

An Entity Relationship Diagram (ER Diagram) is a structural design of the database. It acts as a framework created with specialized symbols for the purpose of defining the relationship between the database entities. ER diagram is created based on three principal components: entities, attributes, and relationships. (Haroon, 2025).

### Multiple entities related to employment tracking:

1. The "Person" entity represents individuals with attributes such as person\_id, name, address, phone, and birth\_date.
2. The "Organization" entity contains details of various organizations, including organization\_id, name, address, phone, and budget\_number.
3. The "Position" entity stores job positions with attributes position\_id, title, and a foreign key linking it to an organization.
4. The "Employment" entity captures relationships between a person and an organization, including employment\_id, employment\_date, termination\_date, and bonus.
5. The "Position\_Held" entity establishes a many-to-many relationship between Person and Position, tracking historical employment records with attributes position\_id, person\_id, start\_date, termination\_date, and salary.
6. The "Permanent\_Employee" and "Part\_Time\_Employee" entities are specializations of "Person."

**Crow’s Foot ER Diagram showing relationships:**



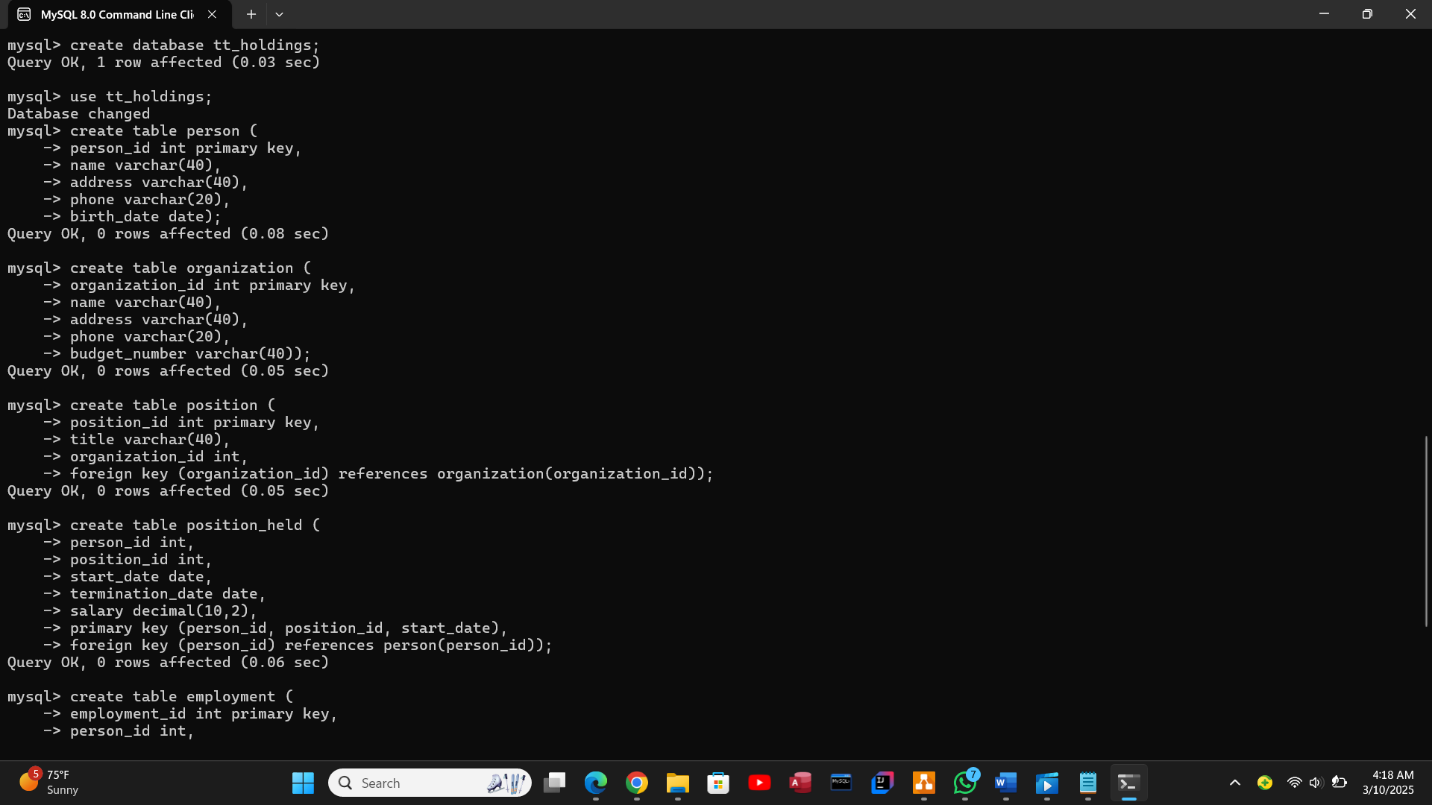
*Figure 1.0 Crow’s foot ERD.*

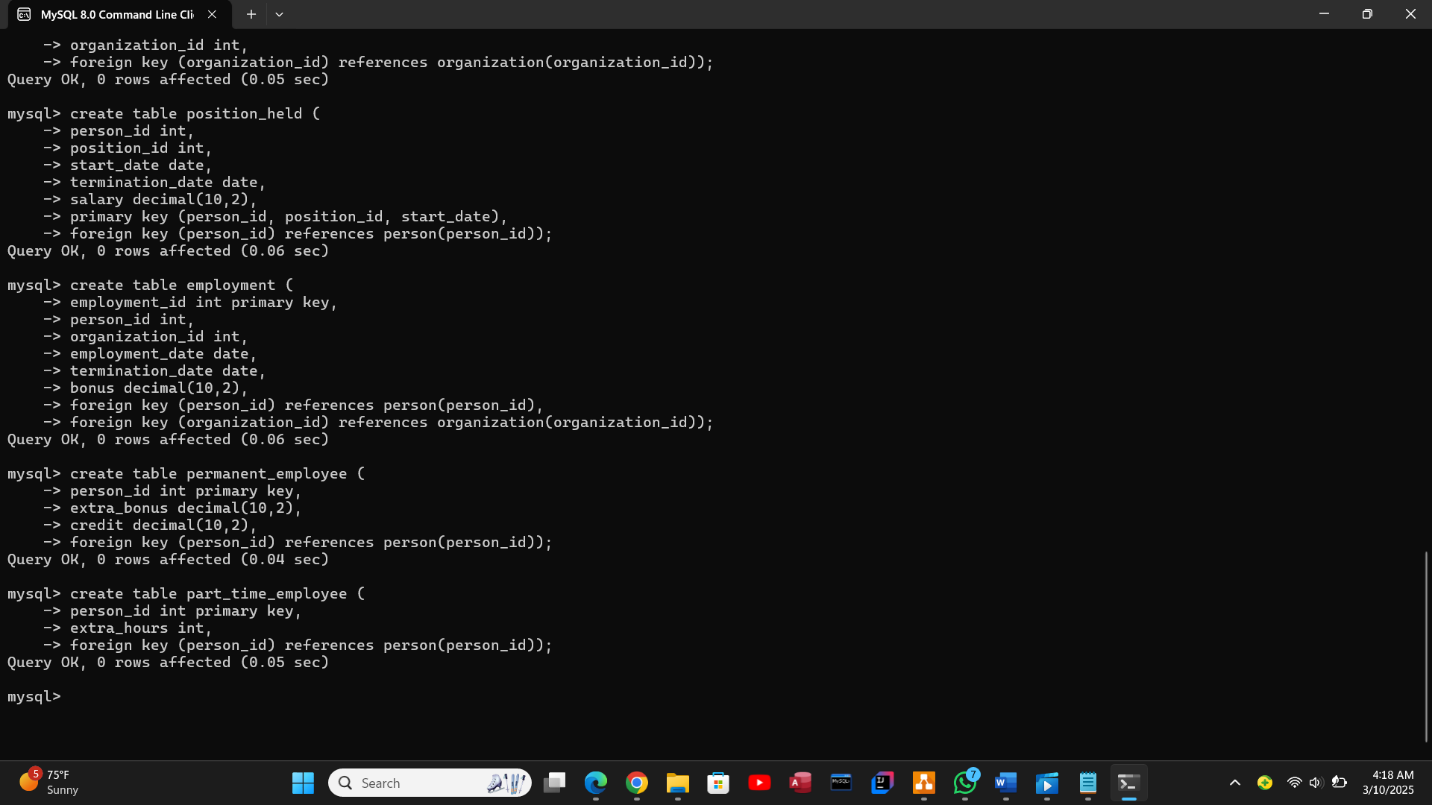
## 3.3 System Implementation / Prototyping

### MySQL database implementation

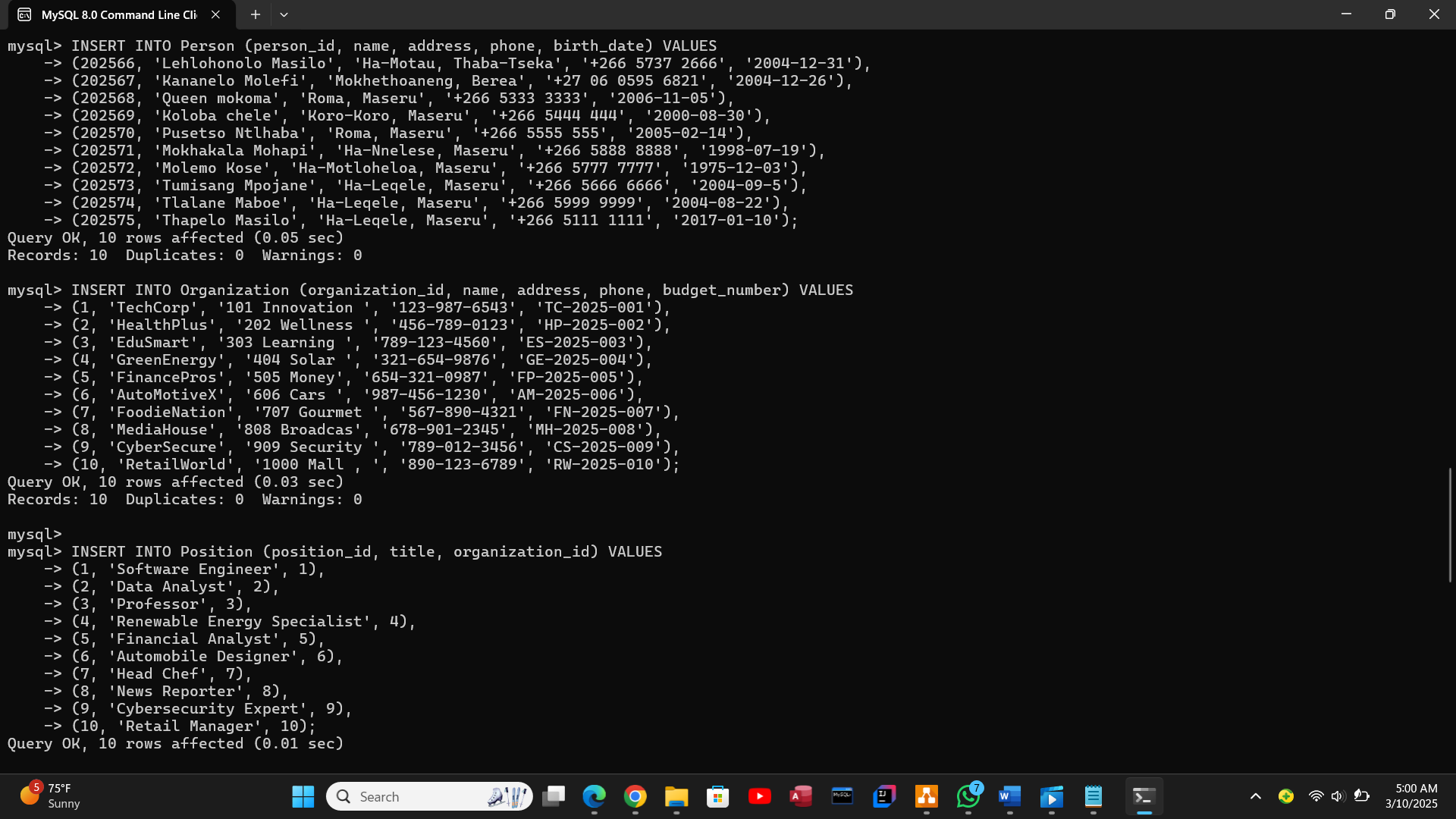
The TT\_HOLDINGS database was created using MySQL, following a relational database model. Incorporating various tables, views, triggers, functions, and stored procedures to ensure data integrity and efficient data retrieval. Below are the steps taken in implementing advanced SQL features.

### Creation of Database and tables

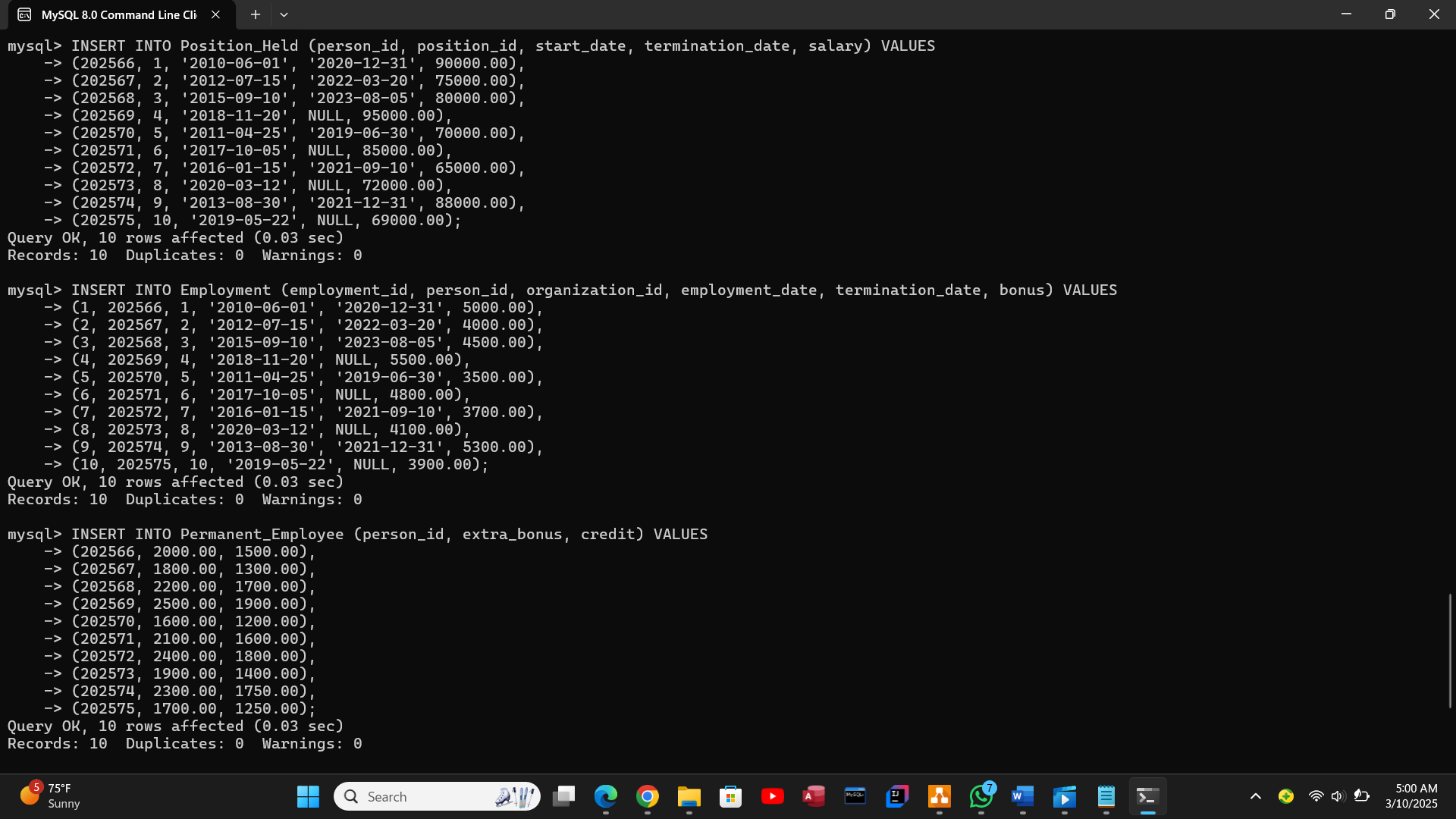


*Figure1.1 Creation of database and first 4 tables.* 

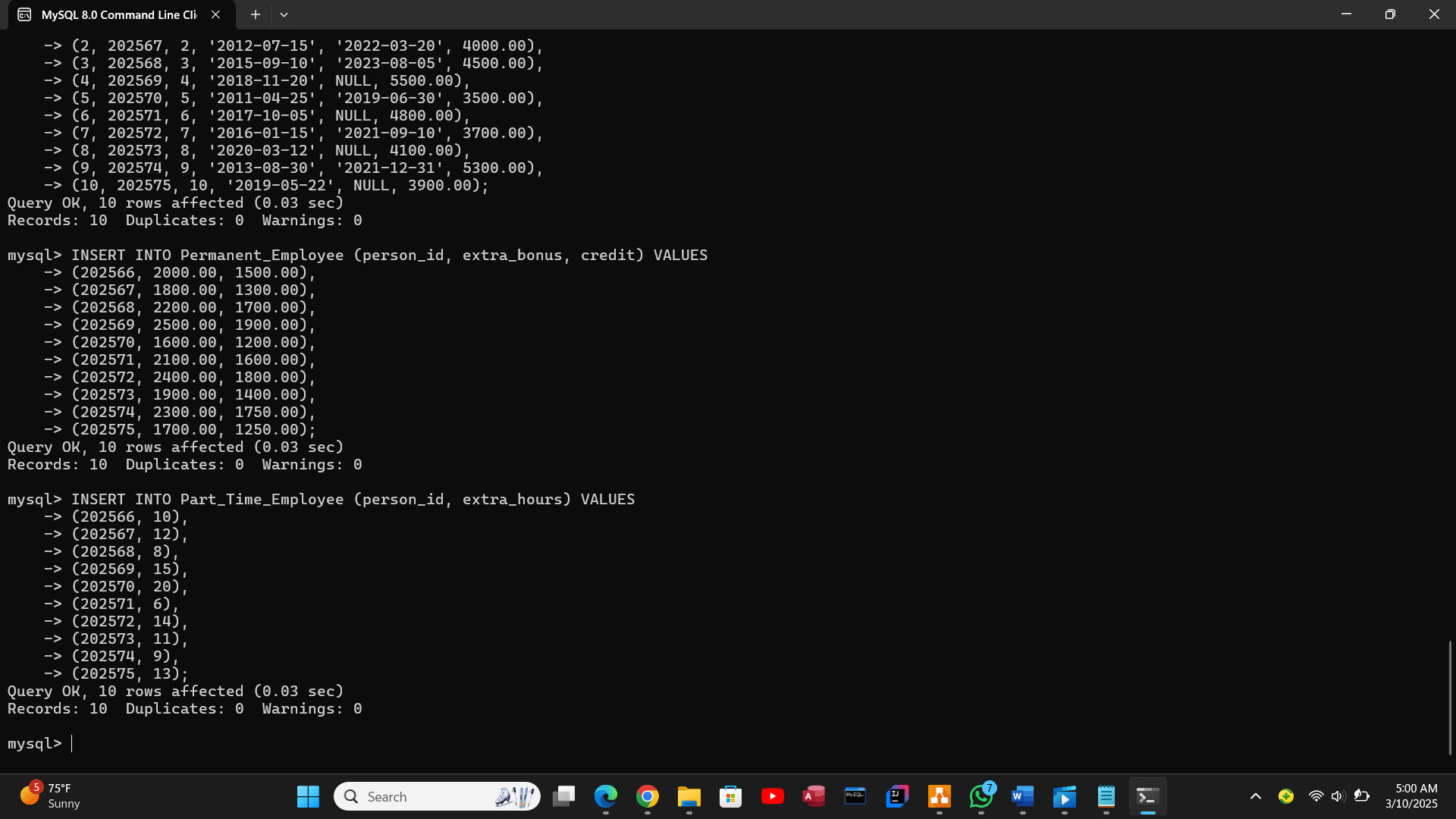
*Figure 1.2 Creation of next 3 tables continued.*

Data insertion

*Figure 1.3 insertion of 10 rows in first 3 tables.*



*Figure 1.4 insertion of next 2 tables.*



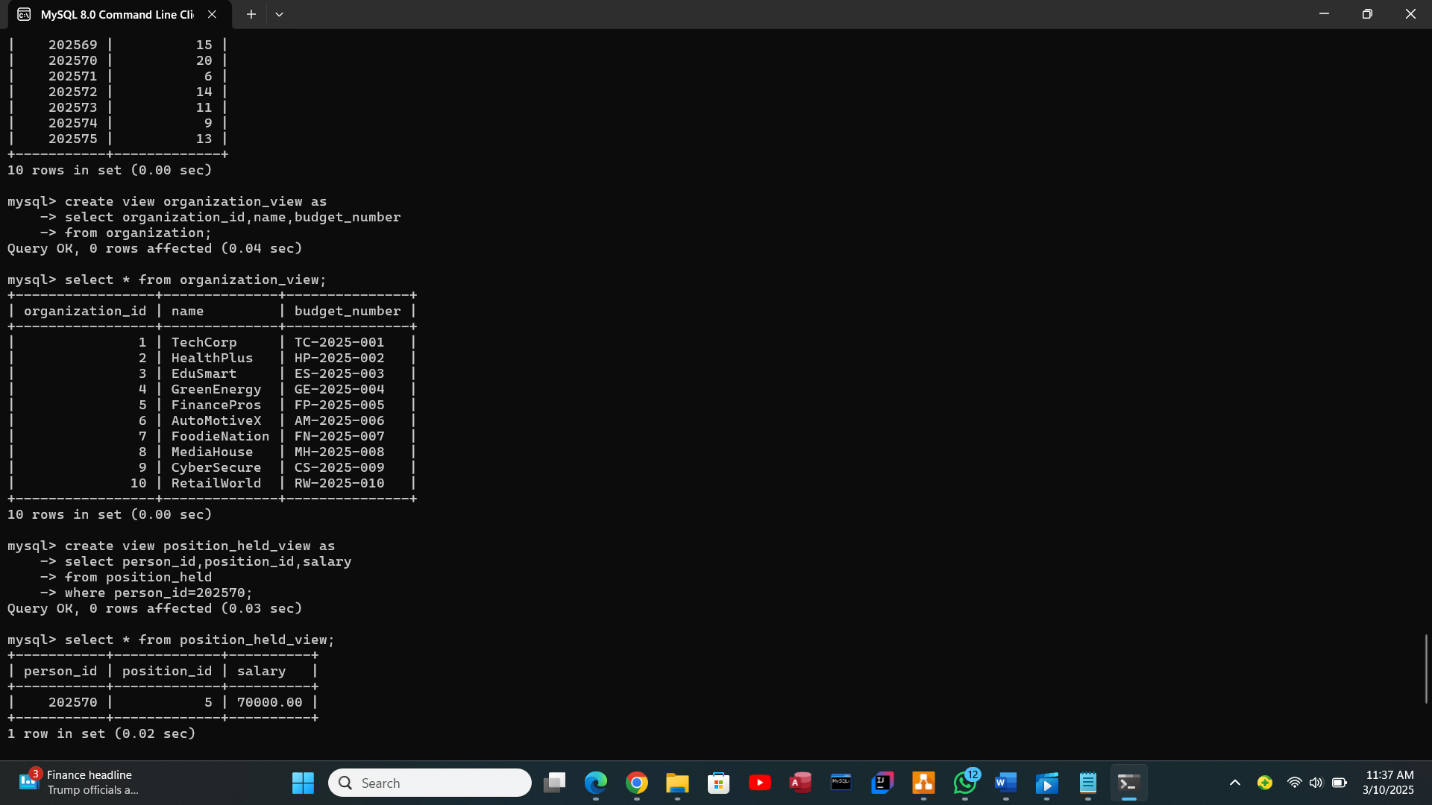
*Figure 1.5 insertion on last 2 tables.*

## Advanced SQL (Views, Triggers, Functions and 2 Procedures).

### Views creation

organization\_view – This view retrieves essential details from the organization table, including organization\_id, name, and budget\_number. It allows users to access organization data without exposing all columns of the table.

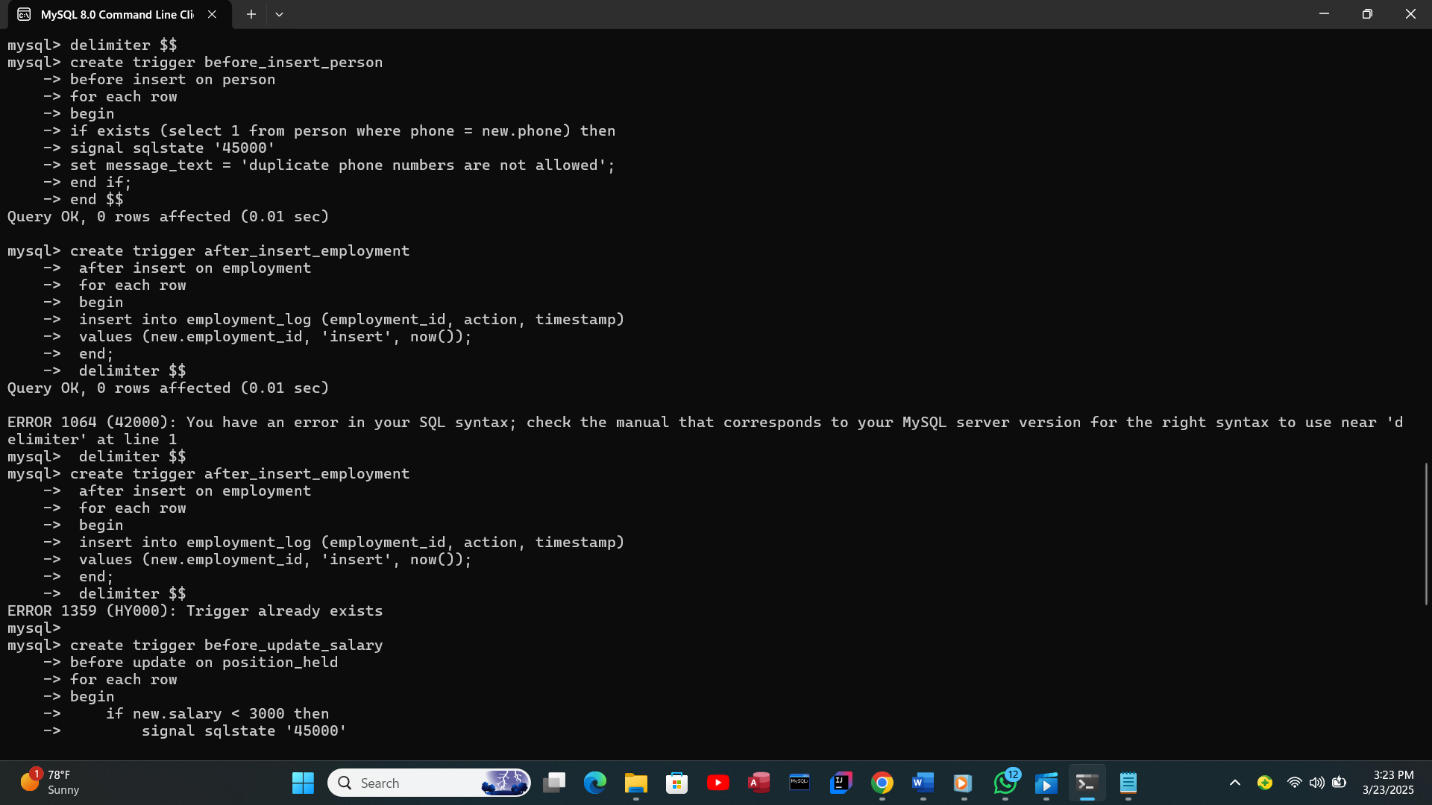
position\_held\_view – This view filters records from the position\_held table to show only details related to a specific person\_id (202570). It displays person\_id, position\_id, and salary, ensuring restricted access to relevant employee position data.

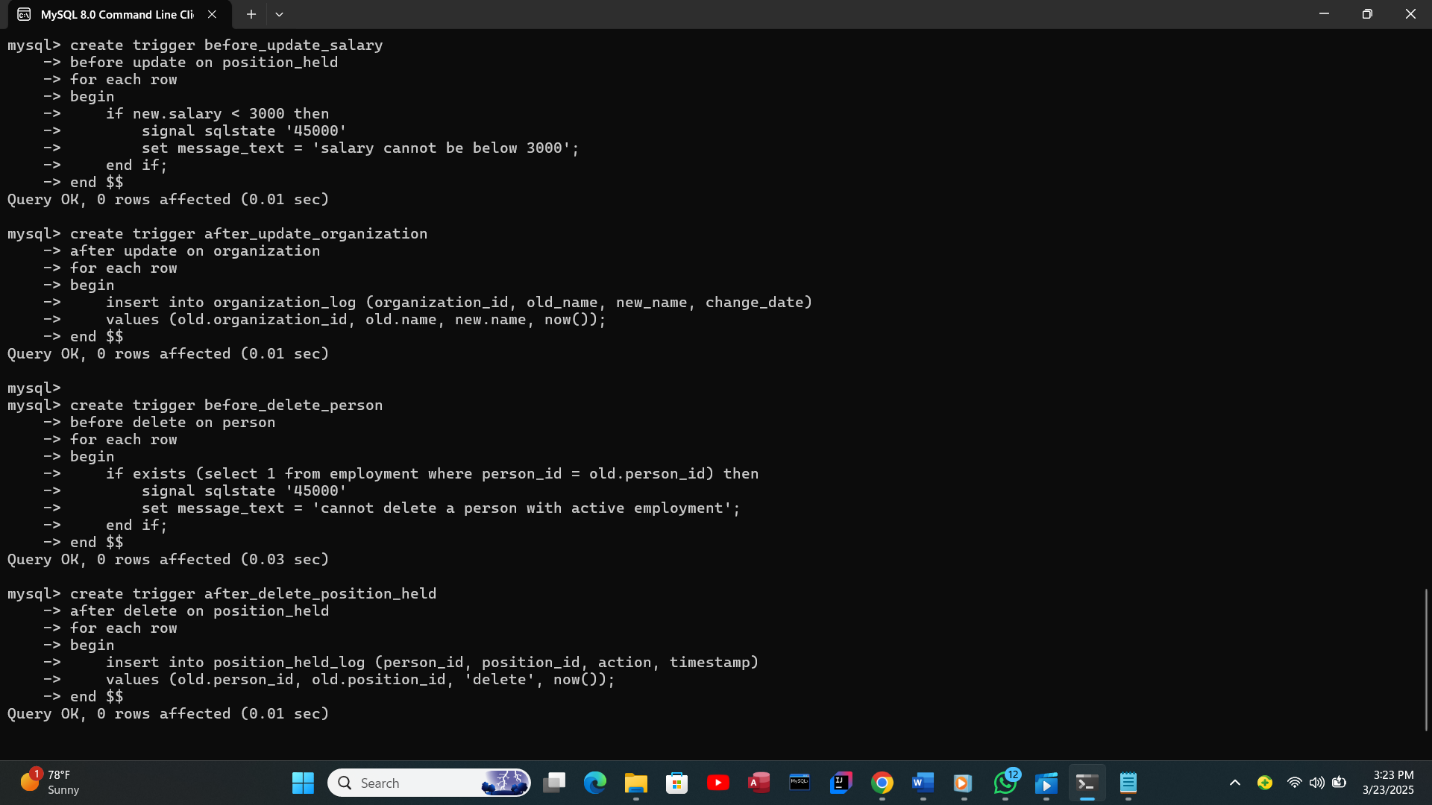


*Figure 2.0 creation of 2 views.*

### Triggers Implementation

These SQL triggers enforce data integrity and maintain logs in the TT Holding Database:

*Figure 3.0 first 2 insert triggers.*

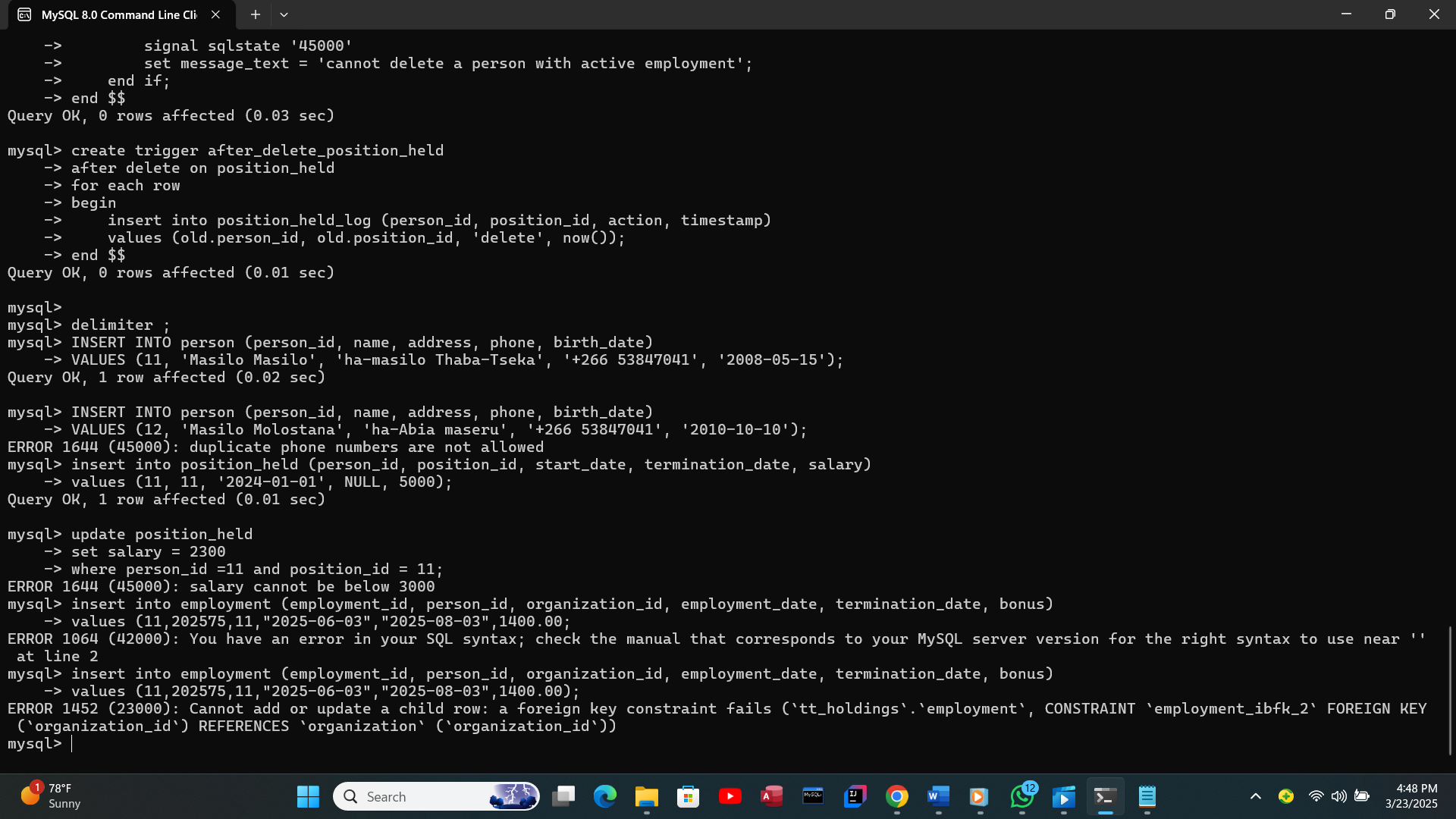
**

*Figure 3.1 4 triggers (2 for update and 2 for delete).*

# 3.4 Testing

Unit testing for individual tables by inserting values into them. Sample data was inserted into each table and data insertion is 10 rows per table.

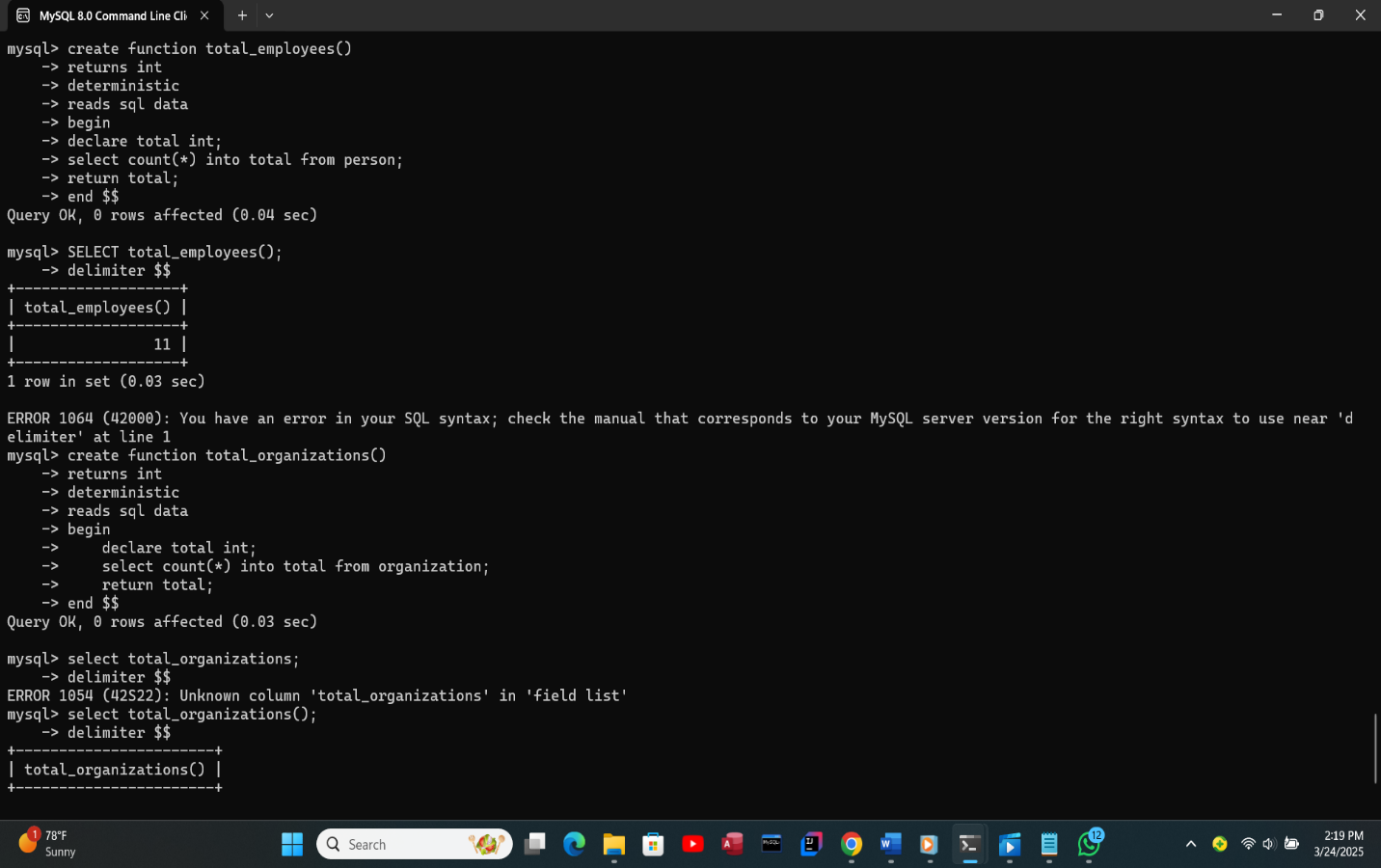
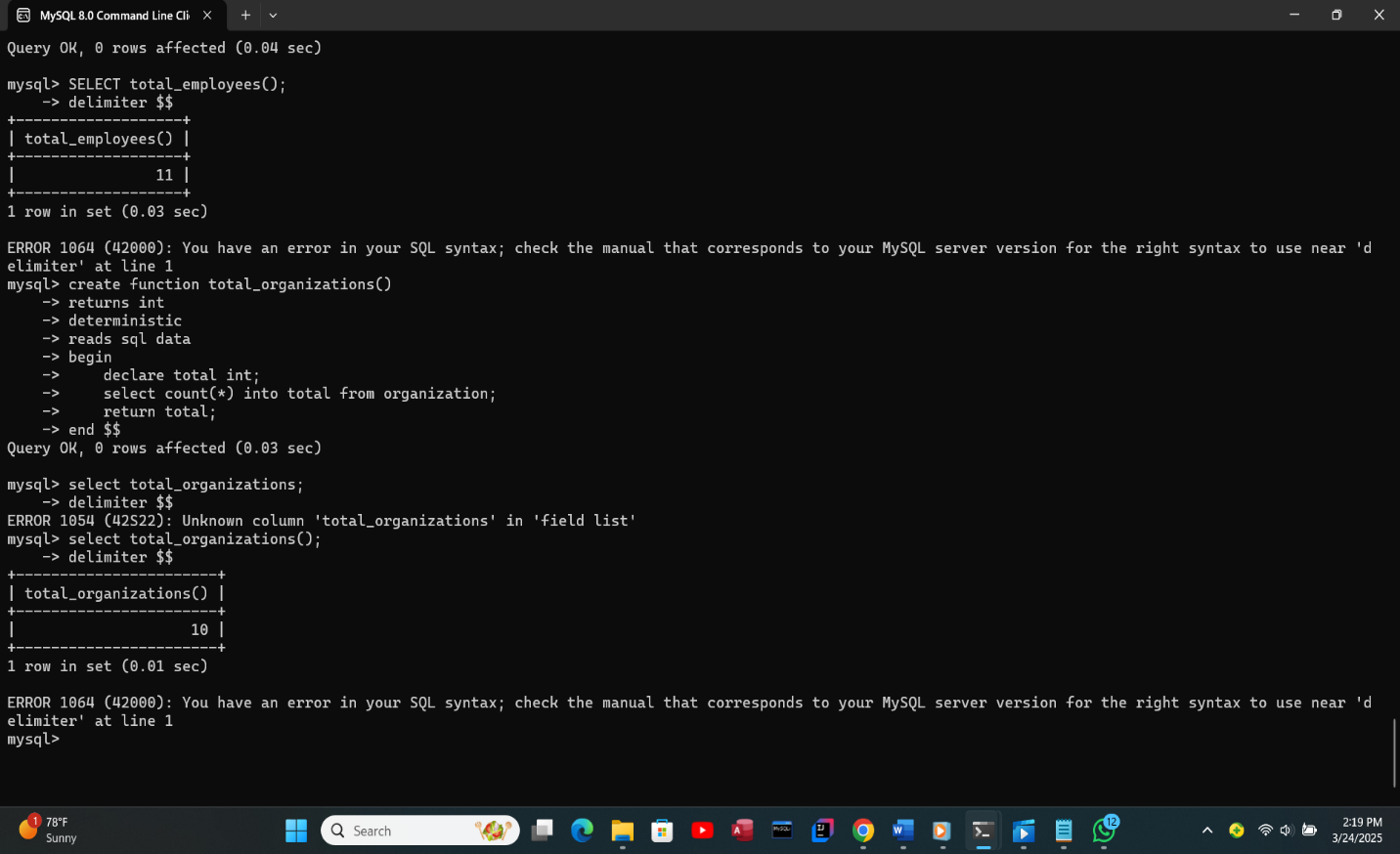
### Test Triggers by Performing Relevant Actions

****

*Figure 3.2 trying to insert a duplicate phone number and Preventing Salary Updates Below a Minimum Value.*

### Functions

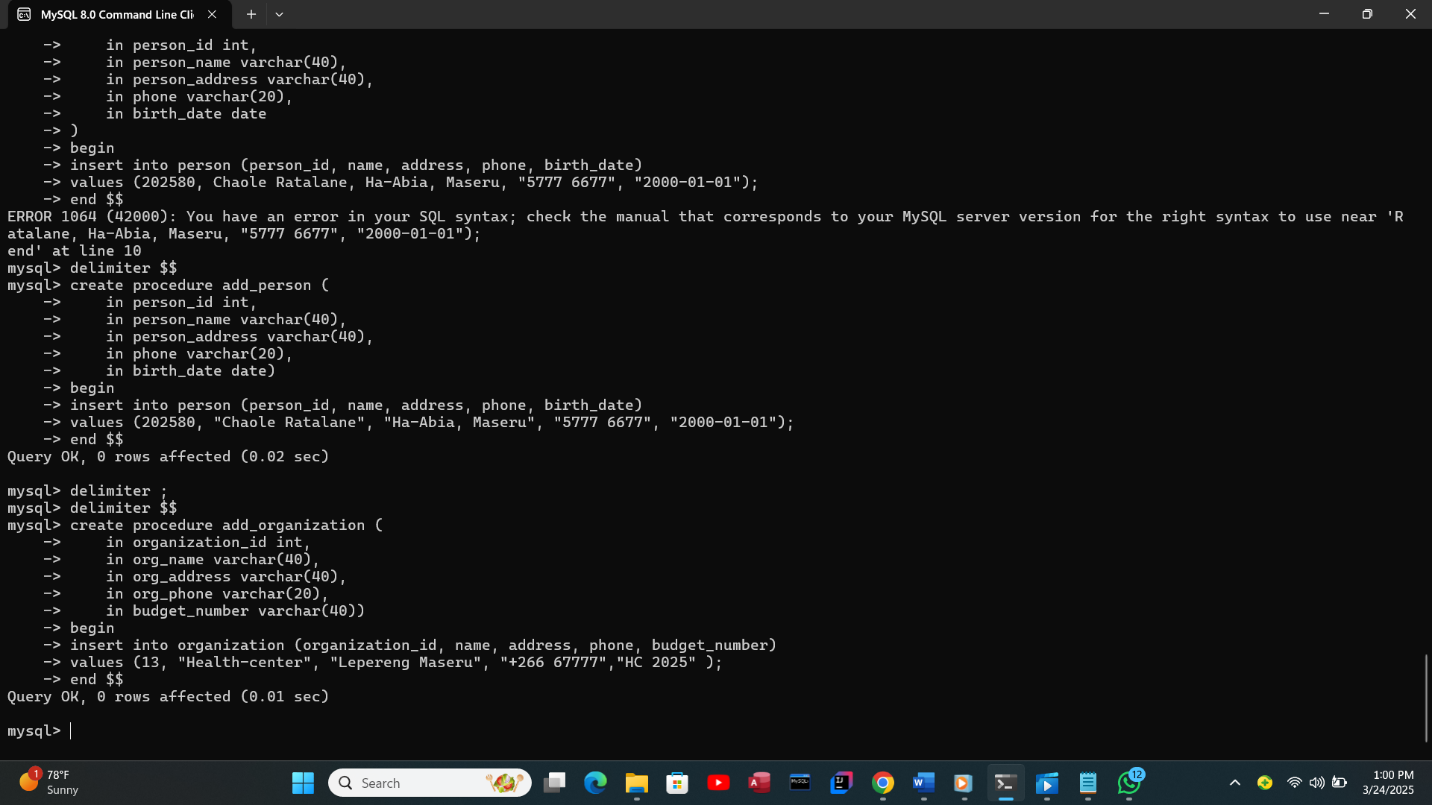
Function in SQL servers are database objects that include a group of SQL statements to perform a specified activity. A function takes parameters, performs actions, and returns the outcome. It should be noted that functions always return either a single value or a table (Shailendra, 2025).

*Figure 4.0 calling the function and get the total number of employees in the person table.*

*Figure 4.1 function getting the total number of organizations in the organization table.*

### Procedures

The add\_person allows the insertion of a new person into the database while ensuring proper data handling and add\_organization procedure simplifies adding organizations by using structured SQL queries.

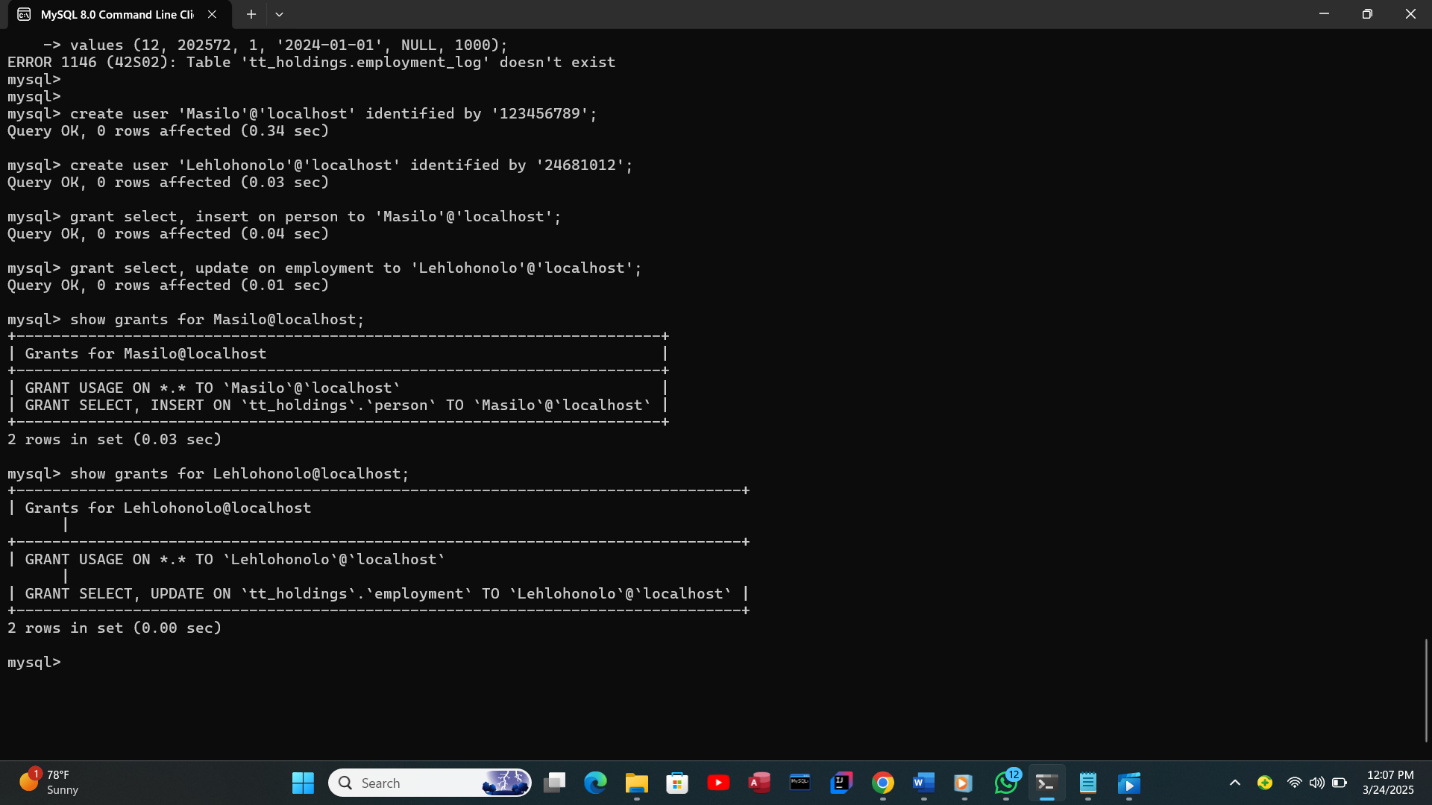


*Figure 5.0 The add\_person procedure and add\_organization procedure.*

### User Creation and Privilege Assignment

SQL users are accounts in a database system that allow authenticated access to resources.  (geeksforgeeks, 2024).

**The SQL commands used for this process are shown below:**



*Figure 5.1 creation of 2 users and granting them privileges.*

# Chapter 4: System Initiation and Planning

## 4.1 Assessing Feasibility of the Project:

Technical Feasibility:

* The database is designed using MySQL, a widely supported relational database management system (RDBMS).
* The system follows a structured schema with relationships between Person, Organization, Employment, and Position tables.

Economic Feasibility:

* MySQL is open-source, eliminating licensing costs.
* Hardware and storage requirements are minimal for small to medium-sized organizations.

Operational Feasibility:

* The database structure supports efficient employee and organization tracking.
* The system is user-friendly with simple SQL queries for data management.

1. **Resource used**
   1. Software Tools: MySQL, ER Diagram tool (Draw.io).

## 4.2 Project Plan

**Development Timeline and Milestones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Tasks** | **Duration** | **Milestone** |
| **Planning & Analysis** | Define requirements, feasibility study | 1 day | Requirements finalized |
| **Design** | Create ERD, relational schema, and mappings | 2 hours | ERD and schema completed |
| **Implementation** | Create tables, relationships, and constraints | 2 days | Database structure completed |
| **Data Insertion** | Insert test data (10-20 rows per table) | 1,5 days | Data successfully inserted |
| **Advanced SQL Development** | Create views, triggers, functions, and procedures | 1 day | SQL components implemented |

# Chapter 5: System Analysis

## 5.1 Determining System Requirements

### Identifying Key Data Entities

* **Person** – Stores employee details.
* **Organization** – Represents companies or institutions.
* **Position** – Defines job roles within an organization.
* **Position\_Held** – Records employment history for each person.
* **Employment** – Tracks employment details between a person and an organization.
* **Permanent\_Employee** – Specialized table for permanent employees.
* **Part\_Time\_Employee** – Specialized table for part-time employees.

**Functional Requirements**

* The system should allow adding, updating, and deleting employee records.
* It should maintain a history of employees’ positions and salaries.
* It must support different employment types (permanent and part-time).

## 5.2 Structuring System Requirements

**Database Schema and Relationships**

* Each table has a primary key for unique identification.
* Foreign keys establish relationships between tables.

# Chapter 6: Conclusion

## 6.1 The Advantages of the System

* Efficient tracking of employment relationships.
* Data consistency and integrity.

## 6.2 The Future Enhancement of the System

* Integration with external HR systems.
* Development of a user-friendly interface.

## 6.3 The Potential Benefits

* Improved workforce management.
* Streamlined database operations.

## 6.4 Conclusion

This report presents the design, implementation, and testing of the TT Holding Database System, which effectively manages employment relationships, positions, and organizations. Through the use of MySQL, advanced SQL features such as triggers, stored procedures, functions, and views were incorporated to ensure data integrity, security, and efficient data management. The system facilitates the management of employee data across multiple organizations, whether permanent or part-time, and tracks the employment history, positions held, and other relevant details.

By employing an Entity-Relationship Diagram (ERD) and normalizing the database, we have created a well-structured and scalable solution. The database supports various advanced SQL techniques, ensuring that actions such as inserts, updates, and deletes are automated and validated through triggers, and that data retrieval is simplified through views.

# References

Alexander, G., 2024. *database (DB).* [Online]   
Available at: https://www.techtarget.com/searchdatamanagement/definition/database  
[Accessed 20 March 2025].

Francois, A., 2024. *Integrity Constraints in SQL: A Guide With Examples.* [Online]   
Available at: https://www.datacamp.com/tutorial/integrity-constraints-sql  
[Accessed 20 March 2025].

geeksforgeeks, 2024. *SQL CREATE USERS.* [Online]   
Available at: https://www.geeksforgeeks.org/sql-create-users/  
[Accessed 21 March 2025].

Haroon, K., 2025. *ER Diagrams in DBMS: Entity Relationship Diagram Model.* [Online]   
Available at: https://www.simplilearn.com/tutorials/sql-tutorial/er-diagram-in-dbms  
[Accessed 20 March 2025].

Shailendra, C., 2025. *Different Types of SQL Server Functions.* [Online]   
Available at: https://www.scholarhat.com/tutorial/sqlserver/different-types-of-sql-server-functions  
[Accessed 21 March 2025].

Shamkani, N., 2020. *FUNDAMENTALS OF DATABASE SYSTEMS.* 7th ed. New York: Ramez, Elmarsri.

# Appendices

Appendix 0: SQL Commands

* A notepad file is included, containing all the SQL commands used to create the database, tables, functions, triggers, procedures, and any other commands executed during the project. This file serves as a comprehensive record of all the steps taken to implement the TT Holding Database System.

Appendix 1: SQL Scripts

* Database Creation Script
* Table Creation Scripts
* Data Insertion Queries
* Views Creation Queries
* Trigger Implementation Scripts
* Stored Procedures and Functions

Appendix 2: ER Diagrams and UML Diagrams

* Crow’s Foot ERD
* Employment Relationship UML Diagram

Appendix 3: Test Cases and Validation

* Test cases for data insertion and retrieval
* Testing triggers

Appendix 4: User Creation and Privilege Assignment

* SQL Queries for User Creation
* Privilege Assignment Queries