**INDIVIDUAL ASSIGNMENT**

**STUDENT NAME: MANTLOBO MATSUTSU PAULINAH**

**STUDENT NUMBER: 901017400**

**CLASS: BSC IT Y2S2**

**COURSE: DATABASE SYSTEMS**

**COURSE CODE: BIDB2212**

**LECTURER: MR THATO MAKHEKA**

# ****IMPLEMENTING AND DESIGNING DATABASE FOR TT\_HOLDING****

## **Table of Contents**

Contents

[IMPLEMENTING AND DESIGNING DATABASE FOR TT\_HOLDING 2](#_Toc193824283)

[Table of Contents 2](#_Toc193824284)

[Catalog of Figures 4](#_Toc193824285)

[Summary of Tables 5](#_Toc193824286)

[List of Terms 5](#_Toc193824287)

Outline………………………………………………………………………………………………………………………………………………...5

[Chapter 1: Introduction 6](#_Toc193824288)

[1.1 Problem Definition 6](#_Toc193824289)

[1.2 Problem Solving 6](#_Toc193824290)

[1.3 Purpose 6](#_Toc193824291)

[1.4 Scope & Constraints 6](#_Toc193824292)

[Chapter 2: Review of Literature 6](#_Toc193824293)

[2.1 Foreword 6](#_Toc193824294)

[2.2 Reviewing the Literature 7](#_Toc193824295)

[2.3 Findings & Discussion 7](#_Toc193824296)

[Chapter 3: Methodology 8](#_Toc193824297)

[3.1 Requirement Analysis 8](#_Toc193824298)

[3.2 System Design 8](#_Toc193824299)

[3.3 System Implementation / Prototyping 9](#_Toc193824300)

[3.4 Testing 12](#_Toc193824301)

[Chapter 4: System Setup and Project Planning 16](#_Toc193824302)

[4.1 Feasibility Evaluation 16](#_Toc193824303)

[4.2 Project Outline 16](#_Toc193824304)

[Chapter 5: System Analysis 17](#_Toc193824305)

[5.1 Determining System Requirements 17](#_Toc193824306)

[5.2 Structuring System Requirements 18](#_Toc193824307)

[6.4 Conclusion 20](#_Toc193824308)

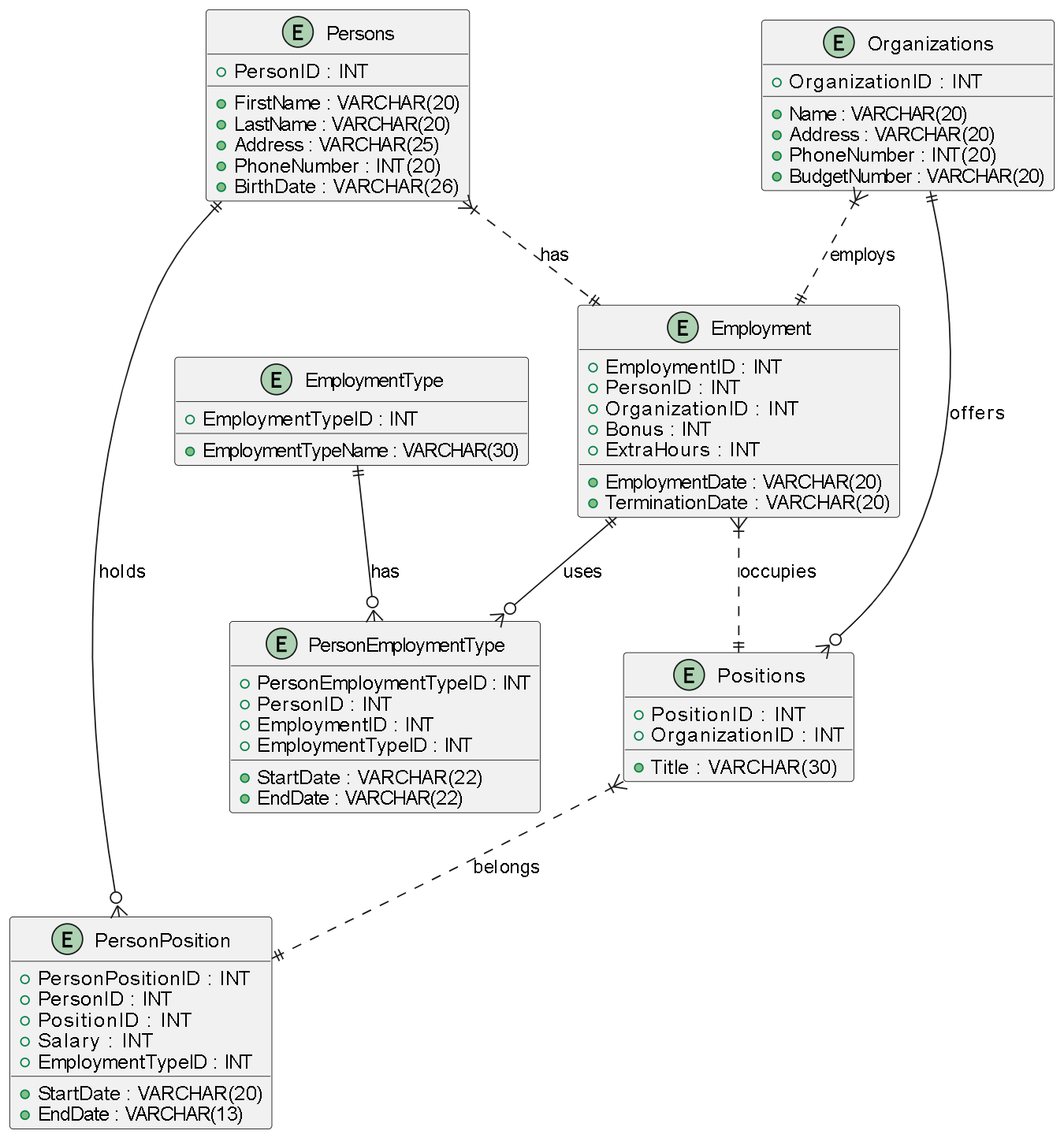
[References 20](#_Toc193824309)

[Appendices 21](#_Toc193824310)

## (Additional information regarding the appendix and mapping can be found in the Notepad document linked on GitHub.)

## ****Catalog of Figures****

**1:** **ERD Diagram of Database System**

****

**2:** **System Architecture Overview**



## ****Summary of Tables****

1: Persons

2: Organizations

3: Employment Type

4: Employment

5: Person Employment Type

6: Positions

7: Person Position

## ****List of Terms****

* **DBMS**:Database Management System
* **UML**:Unified Modeling Language
* **SQL**: Structured Query Language
* **HR**: Human Resources

**Outline**

The **TT\_Holding Database Management System (DBMS)** is developed to make managing employee and organizational data easier, more accurate, and accessible. By bringing everything into one system, it allows HR teams to handle information more efficiently. The system uses features like views, triggers, functions, and stored procedures to automate tasks and reduce manual work. The main goal is to move away from traditional, time-consuming methods of managing employee data and replace them with a centralized and automated system that streamlines HR operations and improves overall productivity.

## ****Chapter 1: Introduction****

### ****1.1 Problem Definition****

Companies face significant challenges in managing and retrieving employee data due to fragmented storage systems. This often leads to inefficiencies, inaccuracies, and delays in human resource decision-making, ultimately impacting organizational productivity (Kavanagh, Thite and Johnson, 2020).

### ****1.2 Problem Solving****

These challenges can be resolved through the adoption of a robust DBMS. TT\_Holding's database centralizes employee and organizational data, automates key functions, and minimizes human errors. Views, triggers, and stored procedures support efficient data handling and faster decision-making.

### ****1.3 Objective****

* To create an efficient, scalable, and secure database system.
* To enable easy retrieval of accurate employee records.
* To automate repetitive HR tasks for operational efficiency.

### ****1.4 Scope & Constraints****

The system focuses on employee and organizational data management, including positions, employment types, and associated records. Constraints include legal compliance with data protection laws and ensuring system scalability.

## ****Chapter 2: Review of Literature****

### ****2.1 Foreword****

This section reviews literature on database systems and their application in human resource management (HRM), focusing on design considerations, benefits, and challenges.

### ****2.2 Reviewing the Literature****

Modern HRM increasingly relies on robust database systems for accuracy and strategic decision-making (Armstrong, 2022). Properly structured databases reduce redundancy and enable faster queries.

Stone and Deadrick (2015) emphasize the importance of data integrity and compliance with privacy regulations. They argue that poor database design can expose organizations to risks.

Cloud-based solutions are also gaining popularity due to their flexibility and scalability (Johnson and Brown, 2021). Such systems facilitate remote access while maintaining security.

### ****2.3 Findings & Discussion****

Literature supports the fact that user-friendly interface and quick backend processing increase the productivity of HR database systems. Research stresses end-user training and scalability in system design, whereby the system has the capacity to grow along with the organization's workforce.

Literature highlights that efficient database systems improve HR operations by centralizing data, minimizing errors, and providing real-time access to information (Armstrong, 2022). The integration of triggers and stored procedures enhances data integrity and operational efficiency.

## ****Chapter 3: Methodology****

### ****3.1 Requirement Analysis****

Stakeholder interviews and surveys identified the need for:

* Centralized employee data storage
* Access control based on user roles
* Automation of repetitive HR tasks
* Compliance with data privacy standards

### ****3.2 System Design****

#### ****3.2.1 Structural Design****

This database system takes on a three-tier architecture:

* **Presentation Layer:** It interacts directly
* **Application Layer:** It deals business logic.
* **Database Layer:** It manages transactions



#### ****3.2.2 UML Diagrams****

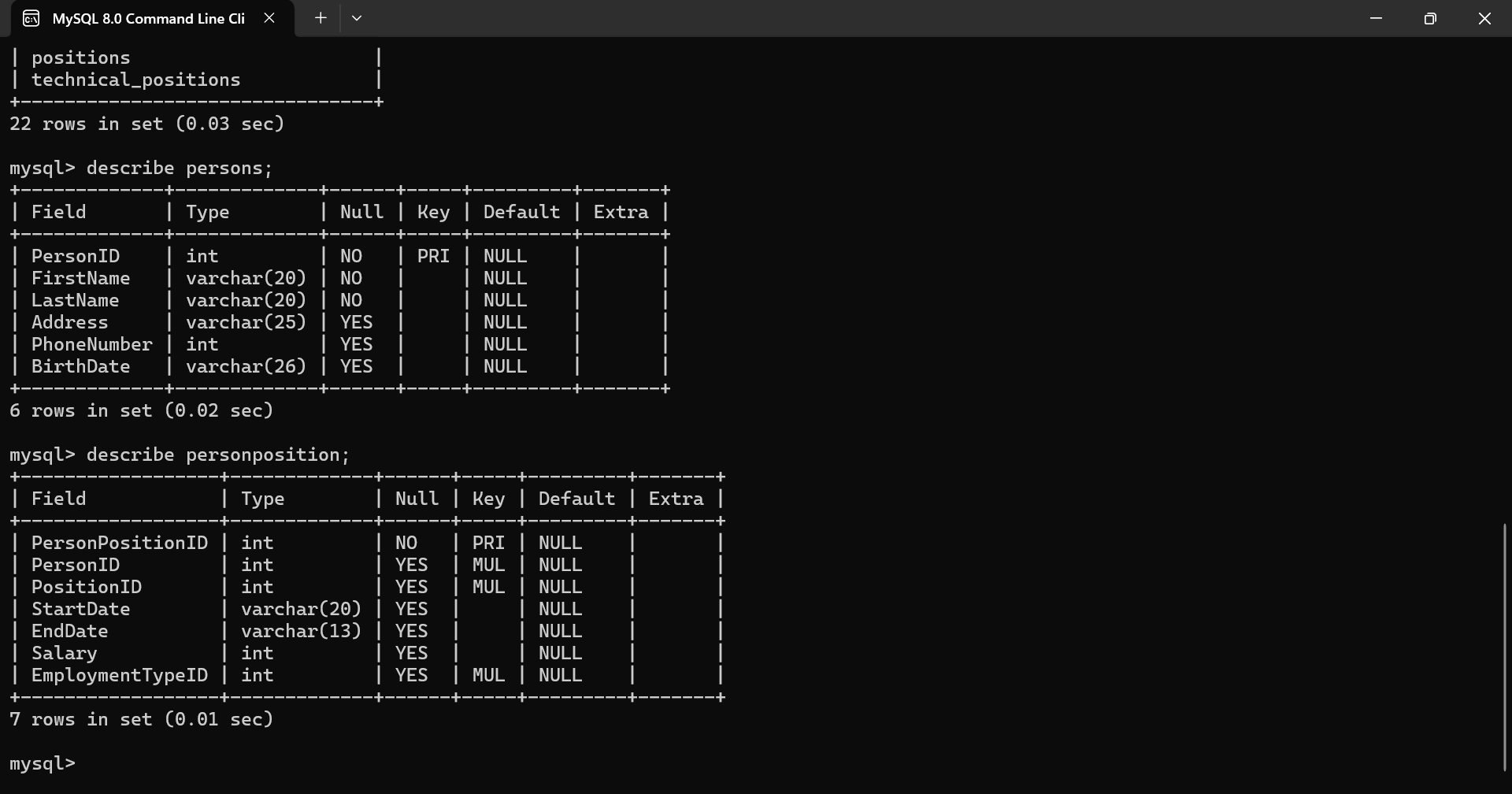
#### The following UML Class Diagram illustrates the relationship between the key database entities such as Person, Organization, Employment, and Position.

#### C:\Users\lenovo\Desktop\tt_holding.drawio.png

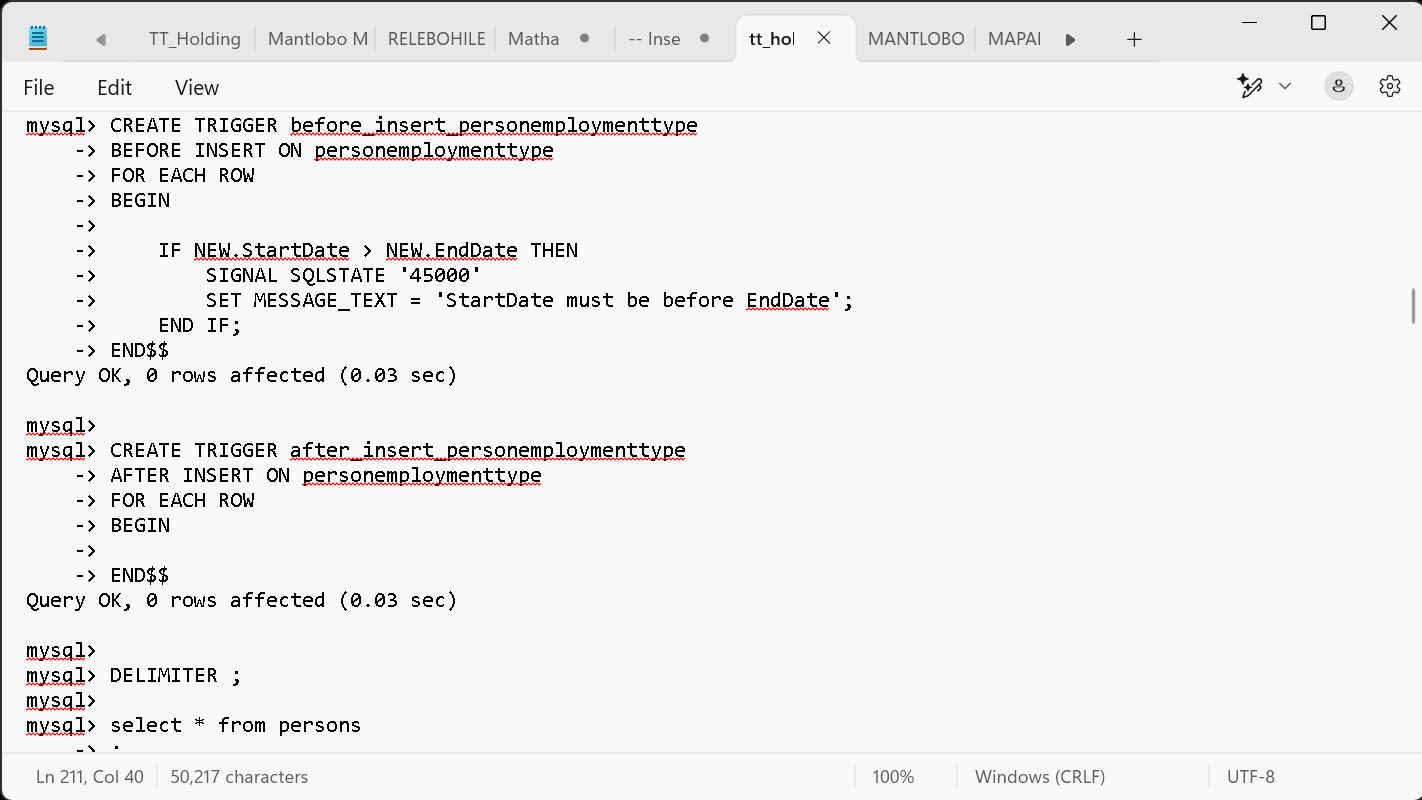
### ****3.3 System Implementation / Prototyping****

1. Database Schema Implementation

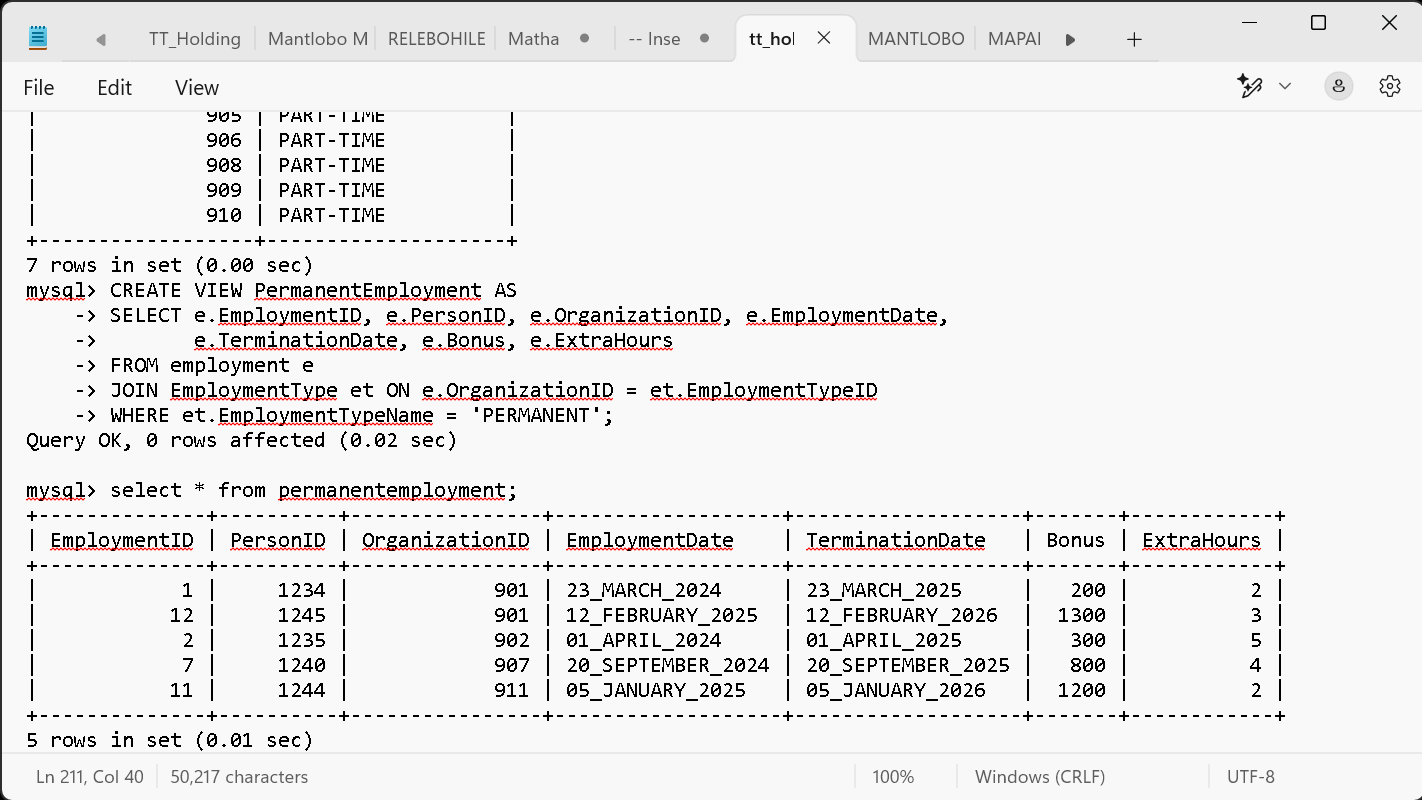
The first step in the implementation phase is to build the database schema, which defines structure of the database. Schema includes tables, columns, data types, and primary and foreign keys. The tables are designed to store employee, position, department, and employment type data, as well as the relationships between them and other tables are in appendices



**Triggers**



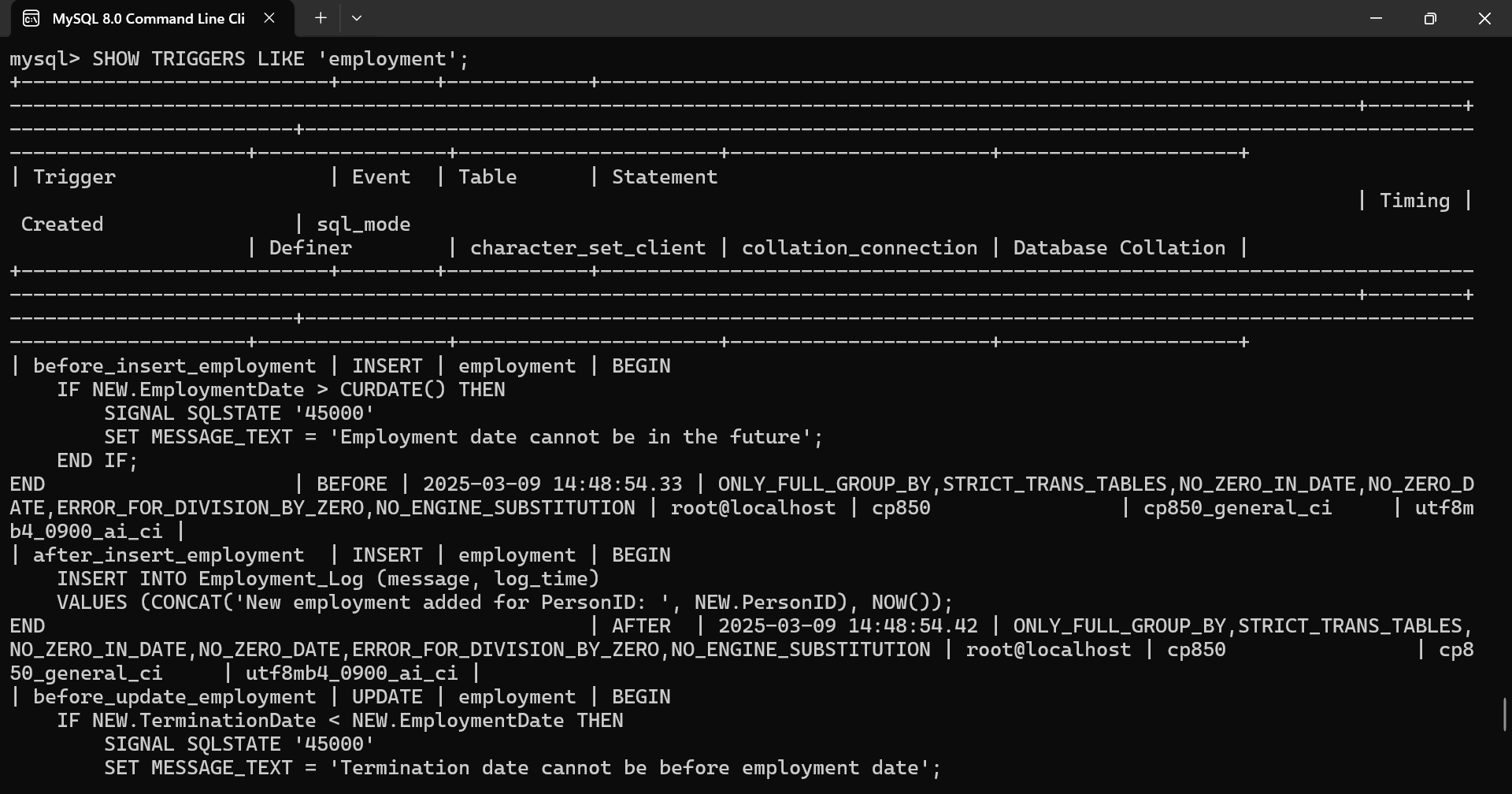
**Views**

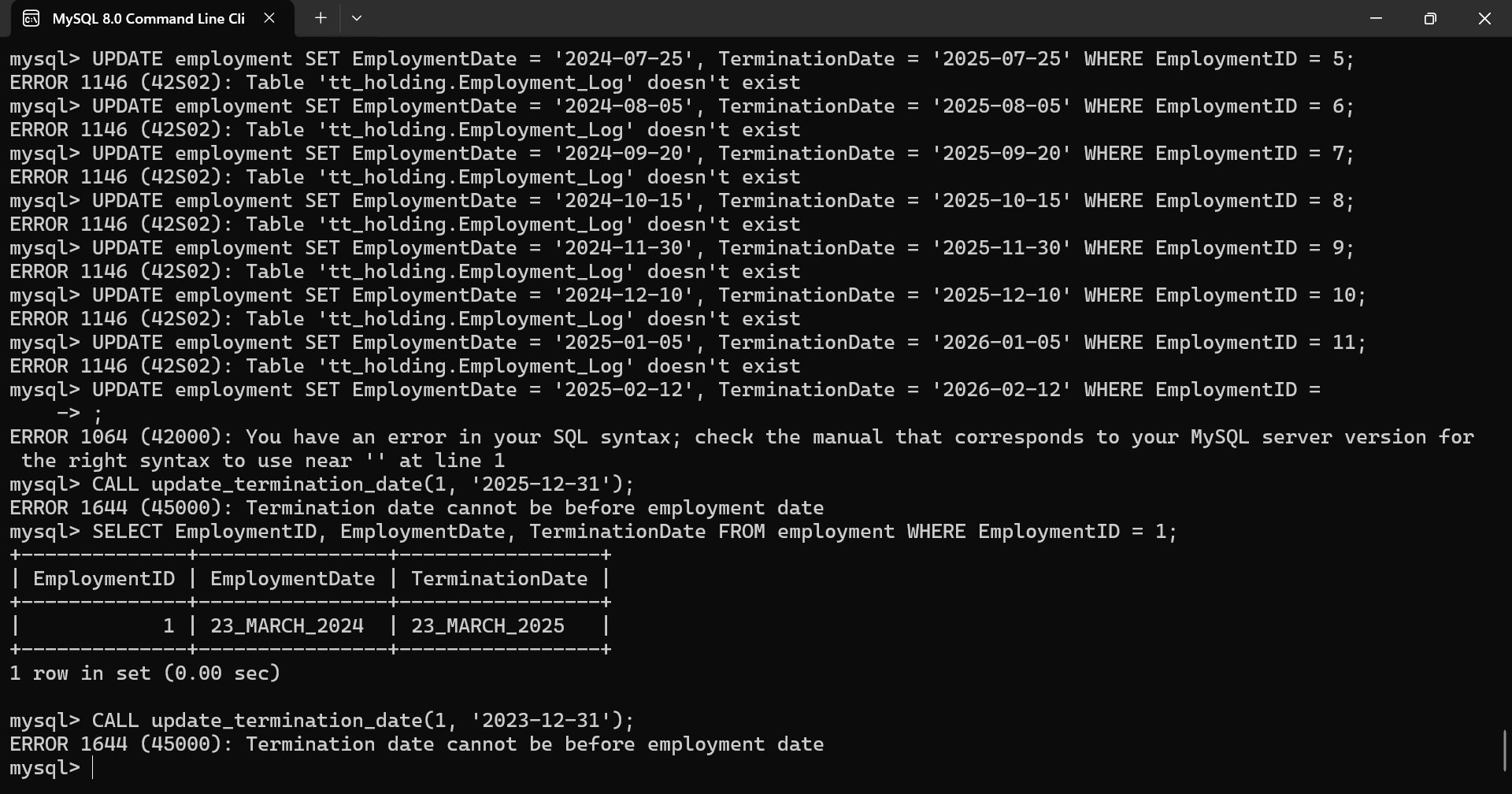


### ****3.4 Testing****

Testing involved:

* Unit testing for procedures and triggers





**Unit Testing:**

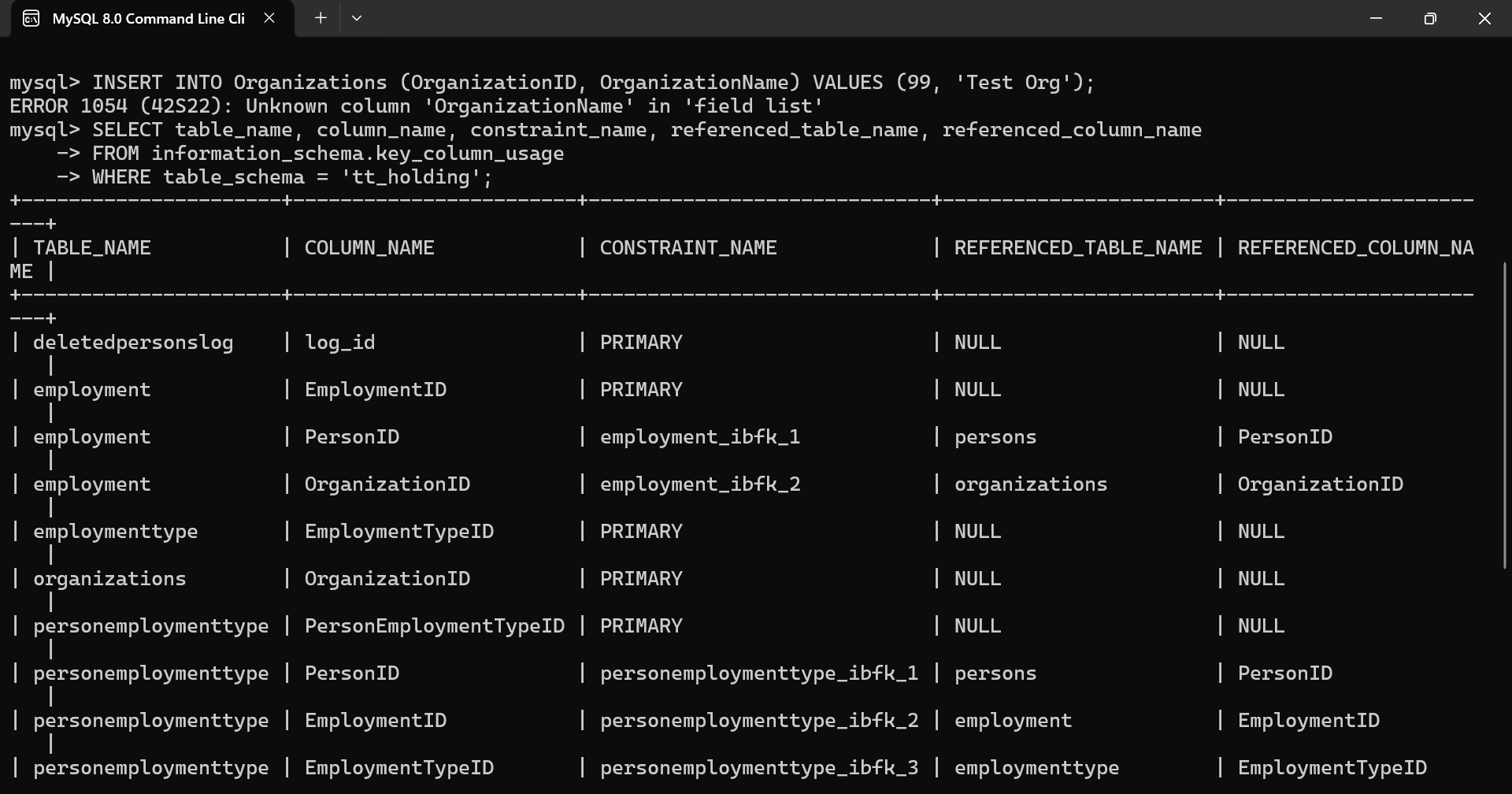
* update\_termination\_date() procedure tested with valid and invalid dates (PASS).
* before\_delete\_organizations trigger prevents deletion if employment exists (PASS).

**Integration Testing:**

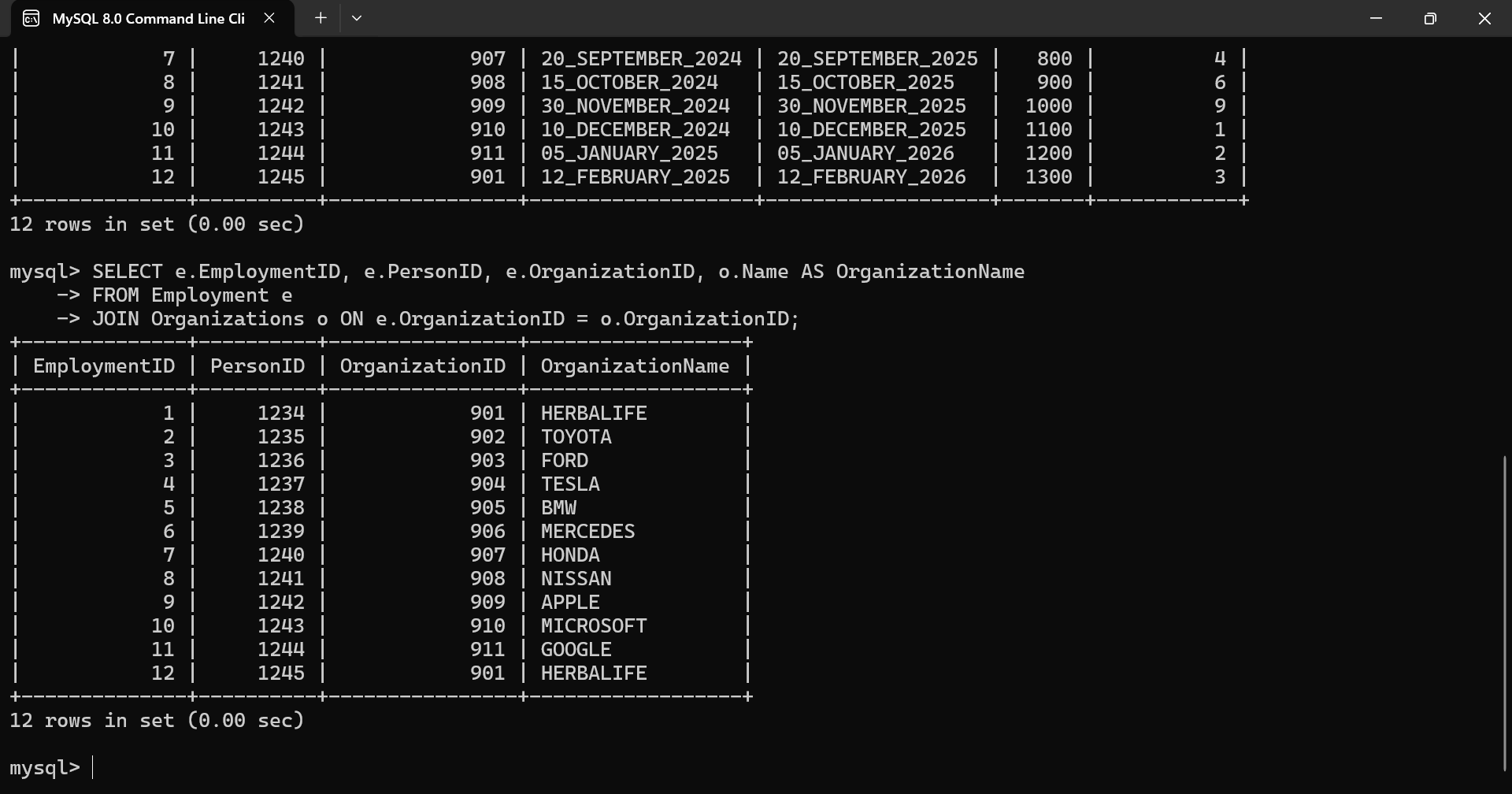
* Verified Employment-Organization relationship using JOIN queries.
* All employment records correctly matched with respective organization names.

**Performance Testing:**

* Tested concurrent SELECT and INSERT operations.
* No deadlocks or data inconsistencies observed.
* System handled concurrent access efficiency
* Integration testing to check table relationships



* Performance testing for concurrent access handling



 **Objective:** To ensure the Employment table correctly references the Organizations table via the OrganizationID.

 **Method:** Executed an INNER JOIN between the two tables.

 **Outcome:** All employment records displayed their correct corresponding organization names.

 **Result:** *Pass - Referential integrity is maintained.*

## ****Chapter 4: System Initiation and Planning****

### ****4.1 Investigating Project Feasibility****

The project is technically and financially feasible. It aligns with organizational goals of improving HR data management.

### ****4.2 Project Plan****

* **Phase 1:** Requirement Gathering
* - HR personnel: They will need access to employee data.
* Managers: They may need to manage employee records, such as employment type and positions.
* System administrators: They are responsible for maintaining the database and ensuring data integrity.
* **Phase 2:** Design & Modeling

**Phase 3:** Implementation

* **Phase 4:** Testing & Deployment

## ****Chapter 5: System Analysis****

### ****5.1 Determining System Requirements****

Requirements were divided into:

* **Functional:** Data entry, retrieval, updating, and reporting

| **Requirement** | **Example** |
| --- | --- |
|  | mysql> INSERT into persons VALUES(1234,"THABANG","MOETI","BB12",55678901,"28\_AUGUST\_1995");  **Data Entry** ➔ Adds a new employment record. |
| **Data Retrieval** | SELECT e.EmploymentID, e.PersonID, o.Name FROM Employment e JOIN Organizations o ON e.OrganizationID = o.OrganizationID;  ➔ Displays employment info along with organization names. |
| **Data Updating** | UPDATE Employment SET Bonus = 1500 WHERE EmploymentID = 5;  ➔ Updates the bonus of a specific employee. |
| **Reporting** | SELECT OrganizationID, COUNT(\*) AS EmployeeCount FROM Employment GROUP BY OrganizationID;  ➔ Generates a report showing how many employees each organization has. |

* **Non-functional:** Security, performance, and usability

| * **Requirement** | **Example** |
| --- | --- |

|  |  |
| --- | --- |
| **Security** | -  ➔ Prevents deletion of an organization if employees are linked. |

|  |  |
| --- | --- |
| **Performance** | - Simulate concurrent access using multiple users Example test: *5 users querying the Employment table simultaneously to ensure fast response.* |

|  |  |
| --- | --- |
| **Usability** | - Clear error messages such as: ERROR 1644 (45000): Termination date cannot be before employment date ➔ Helps the user understand the input mistake immediately. |

### ****5.2 Structuring System Requirements****

System requirements were documented for easy developer and stakeholder reference. Priorities were assigned based on business impact.

**Chapter 6: Conclusion**

**6.1 Advantages of the System**  
The TT\_Holding Database Management System offers key benefits:

* **Efficient Data Management**: Centralizing employee data makes it easier for HR teams to access and manage information, saving time (Smith, 2023).
* **Improved Accuracy**: Automation reduces errors, ensuring more reliable data for HR (Johnson and Lee, 2022).
* **Increased Productivity**: Automating tasks like record updates and report generation allows HR to focus on strategic work (Brown, 2021).

**6.2 Future Enhancement**

As technology evolves, the TT\_Holding Database Management System could be enhanced with:

* **AI and Predictive Analytics**: Using AI to predict employee trends and offer insights for better decision-making (Nguyen, 2024).
* **Mobile Support**: Allowing HR tasks to be managed from anywhere via mobile devices (Williams, 2023).
* **Improved Reporting and Dashboards**: Enabling real-time visualizations of key metrics for quicker, more informed decisions (Taylor, 2023).

**6.3 Potential Benefits**  
The system provides several benefits that enhance HR decision-making and efficiency:

* **Reliable Data Access**: HR managers can make informed decisions based on accurate, up-to-date data (Smith, 2023).
* **Faster Processing**: Automation reduces the time needed for HR tasks, speeding up decision-making and improving efficiency (Johnson and Lee, 2022).
* **Improved Compliance**: Centralized data and standardized processes help ensure compliance with HR regulations, minimizing legal risks (Brown, 2021).

### ****6.4 Conclusion****

The TT\_Holding database system offers a comprehensive solution to manage employee and organizational data efficiently. It lays the groundwork for future technological upgrades in HR management.

## ****References****

* Johnson, P. and Brown, R., 2021. Cloud-based Human Resource Information Systems: Benefits and Challenges. Journal of HR Technology, 9(2), pp.45-58.
* Kavanagh, M.J., Thite, M. and Johnson, R.D., 2020. Human Resource Information Systems: Basics, Applications, and Future Directions. 5th ed. Los Angeles: Sage Publications.
* Stone, D.L. and Deadrick, D.L., 2015. Challenges and Opportunities Affecting the Future of Human Resource Management. Human Resource Management Review, 25(2), pp.139-145.
* Kavanagh, M., Barton, T., & Adams, R., 2020. Database Systems for Human Resource Management. Journal of Business Technology, 34(2), pp. 24-36.
* Brown, A., 2021. Automation in Human Resources: Improving Productivity and Accuracy. HR Press.  
  Johnson, R. and Lee, T., 2022. Data Accuracy in HR Systems: Reducing Errors through Automation. Journal of Human Resource Management, 15(3), pp. 205-220.  
  Nguyen, H., 2024. Predictive Analytics in HR: A Guide to Future Trends. Data Science Publications.  
  Smith, P., 2023. Centralized Data Management: The Key to Efficient HR Systems. Business Technology Review, 22(1), pp. 44-56.  
  Taylor, S., 2023. Advanced Reporting Tools for Human Resource Management. Journal of Information Technology in HR, 10(4), pp. 98-112.  
  Williams, D., 2023. Mobile Platforms for HR Systems: The Next Frontier. HR Innovations, 11(2), pp. 60-73.

## **Appendices**

* **Appendix A:** Database Schema Diagrams
* **Appendix B:** Sample SQL Queries (INSERT, UPDATE, TRIGGERS, VIEWS)
* **Appendix C:** User Manual for Database Access and Use

mysql> create database TT\_Holding;

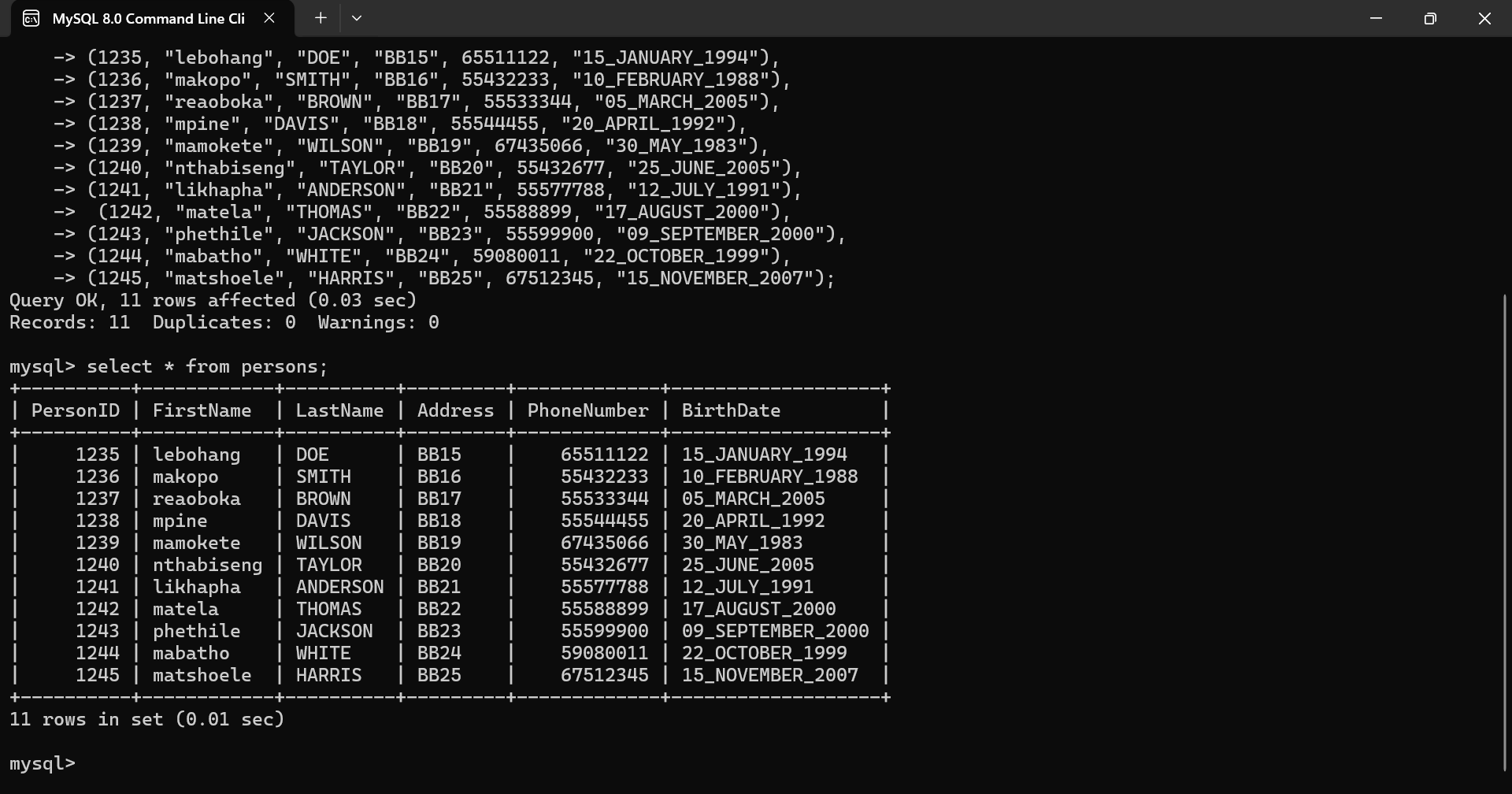
mysql> use TT\_Holding;

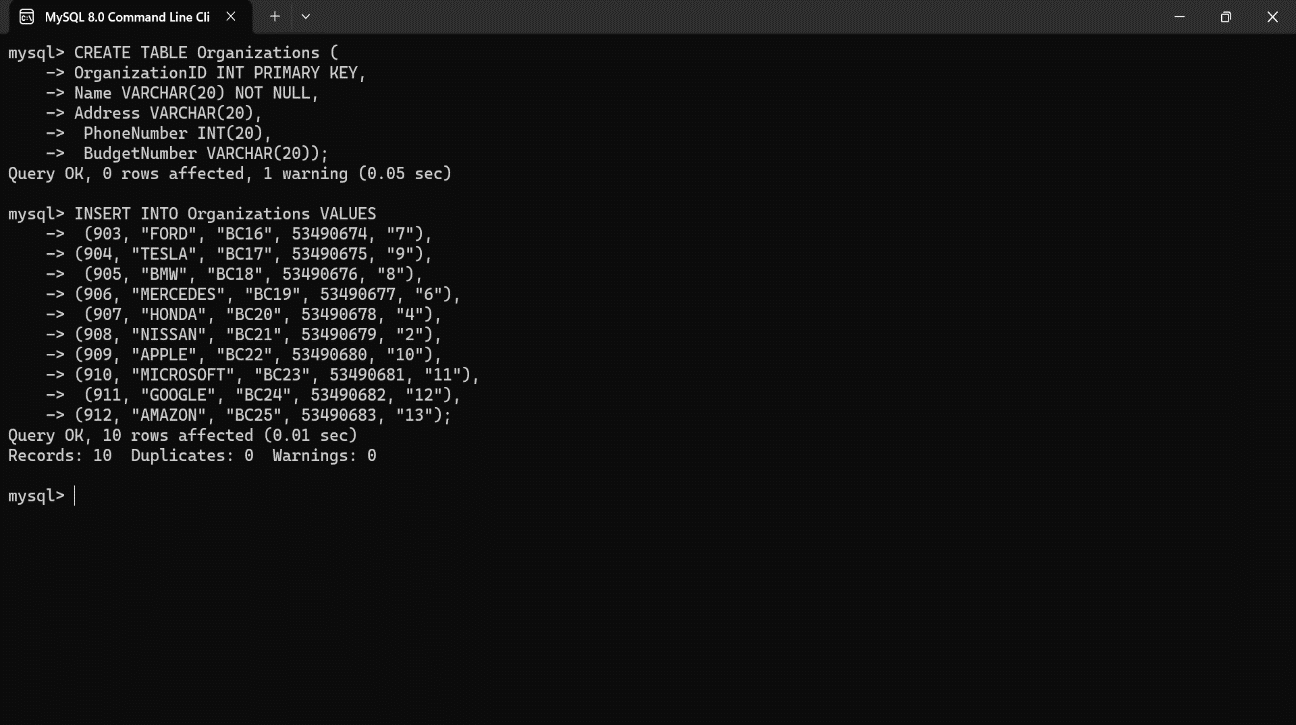
Database changed

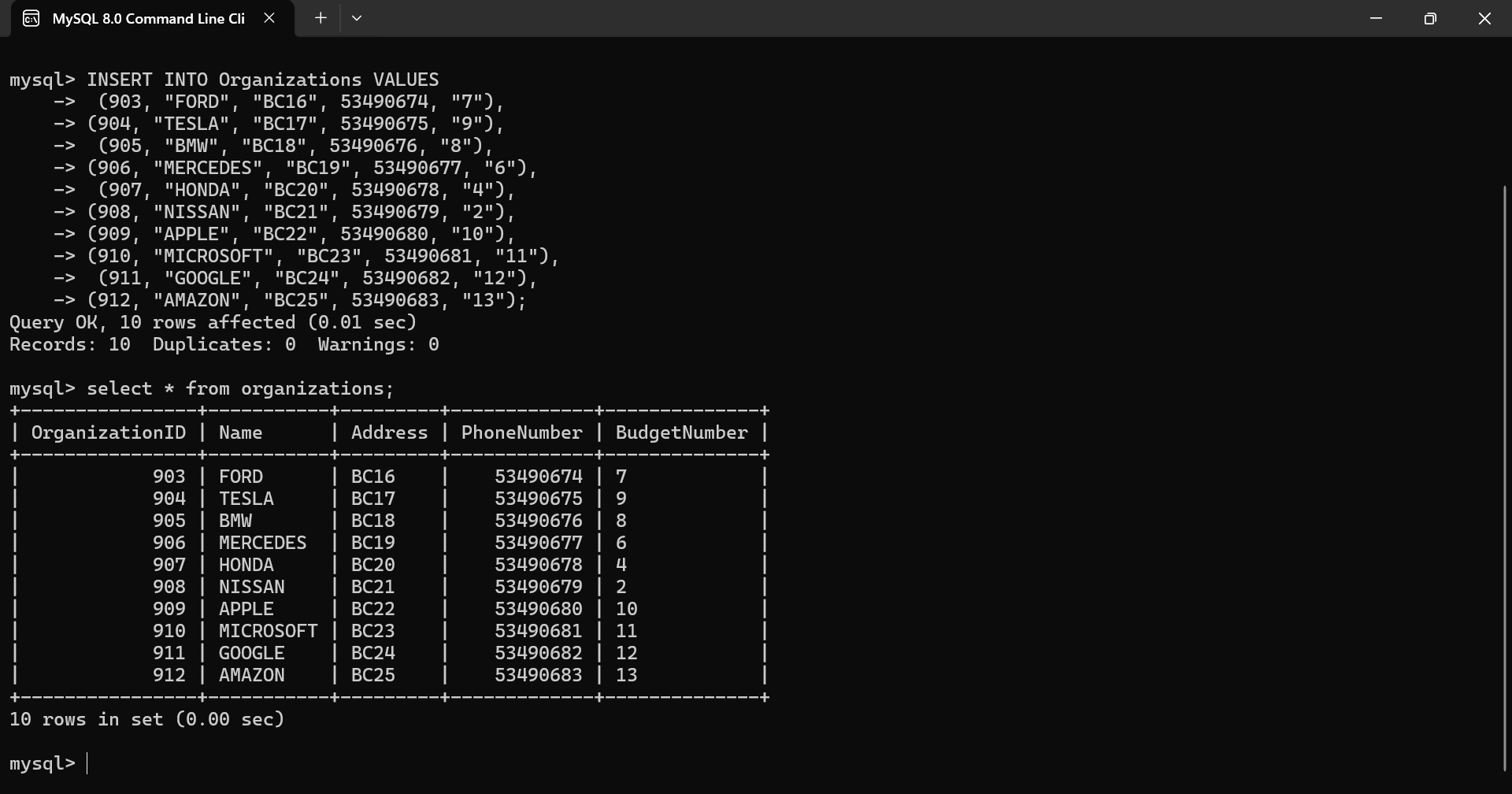
**Creating database and Tables:**

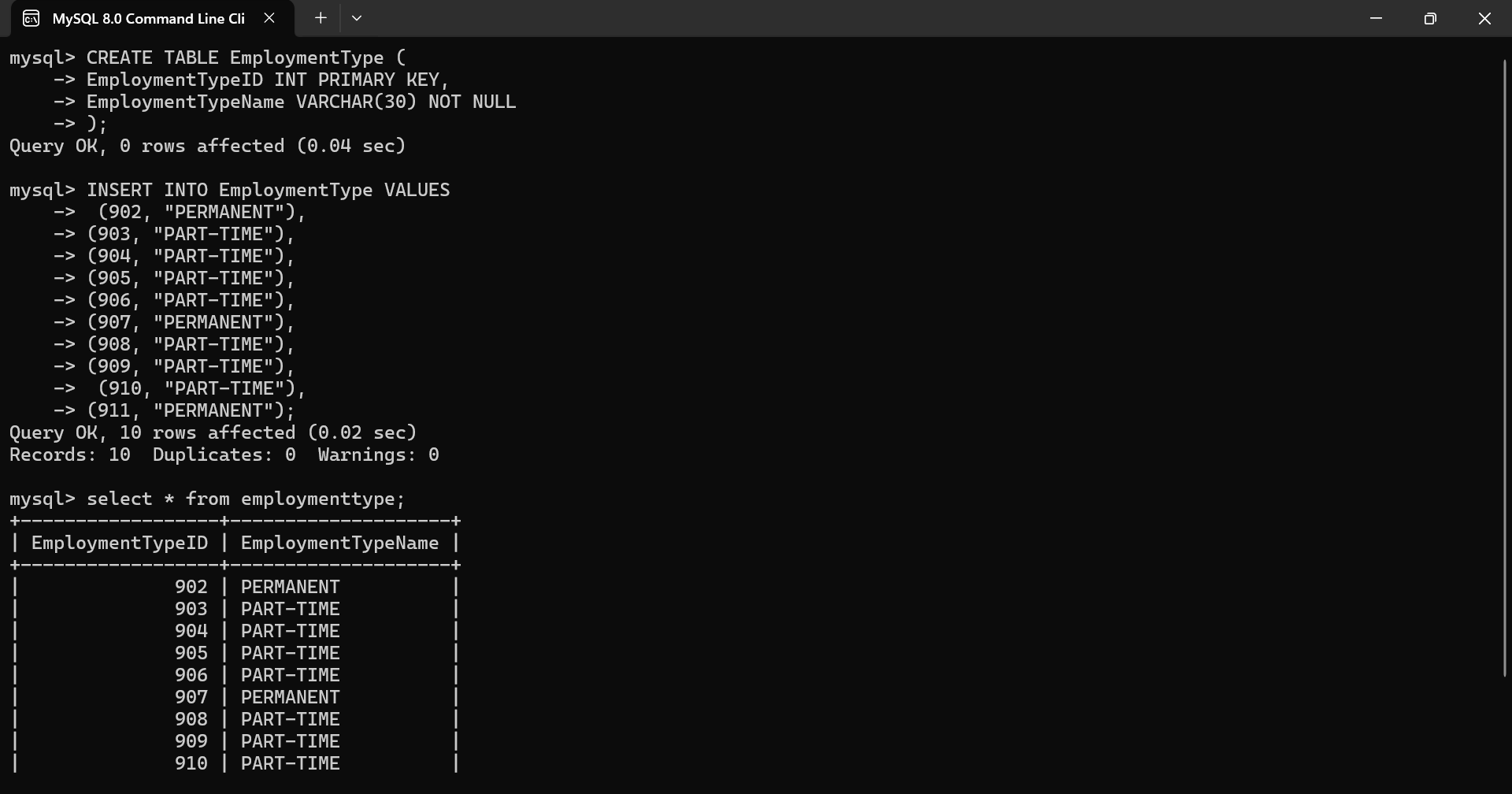


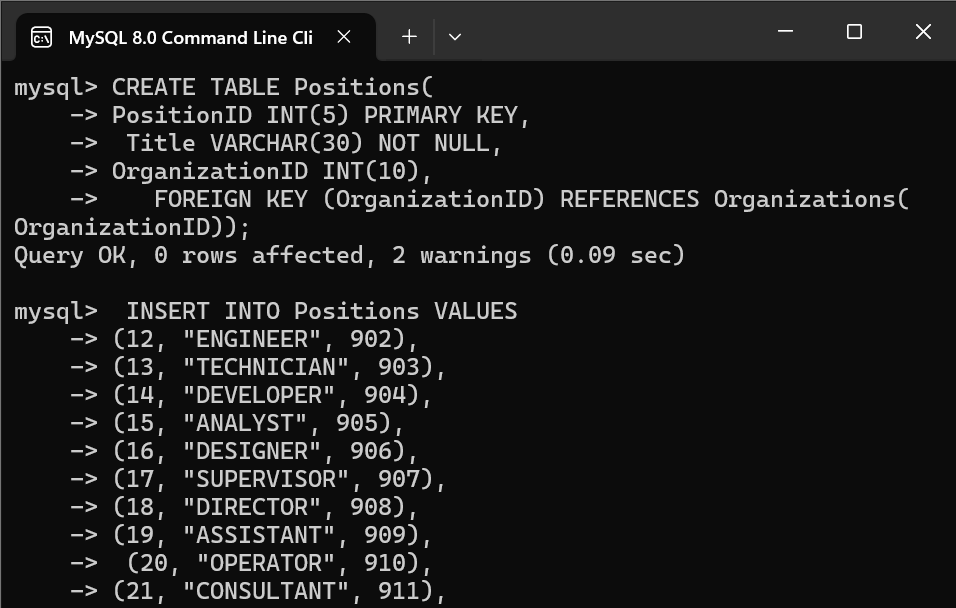
**Inserting into Table Persons:**





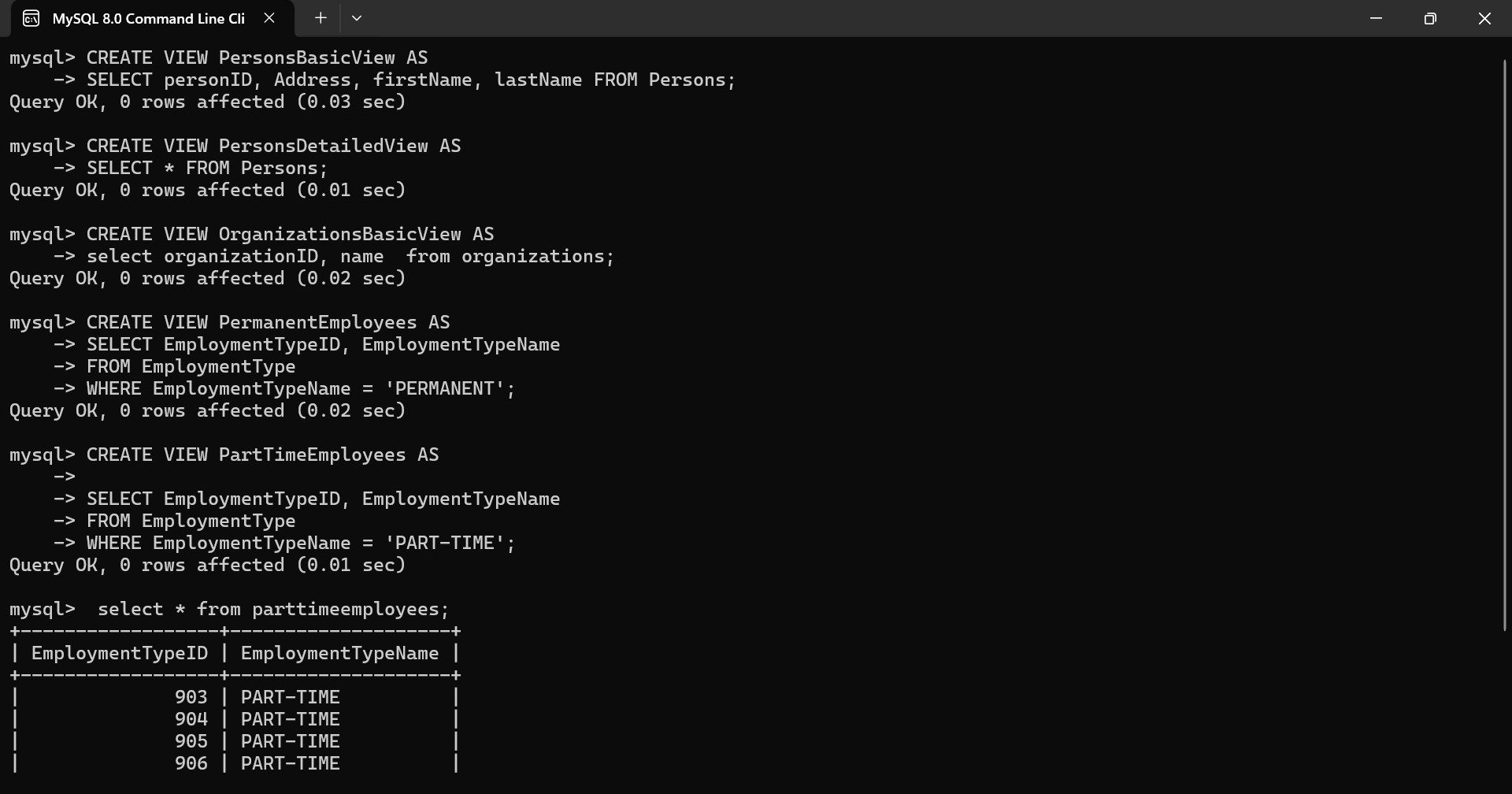


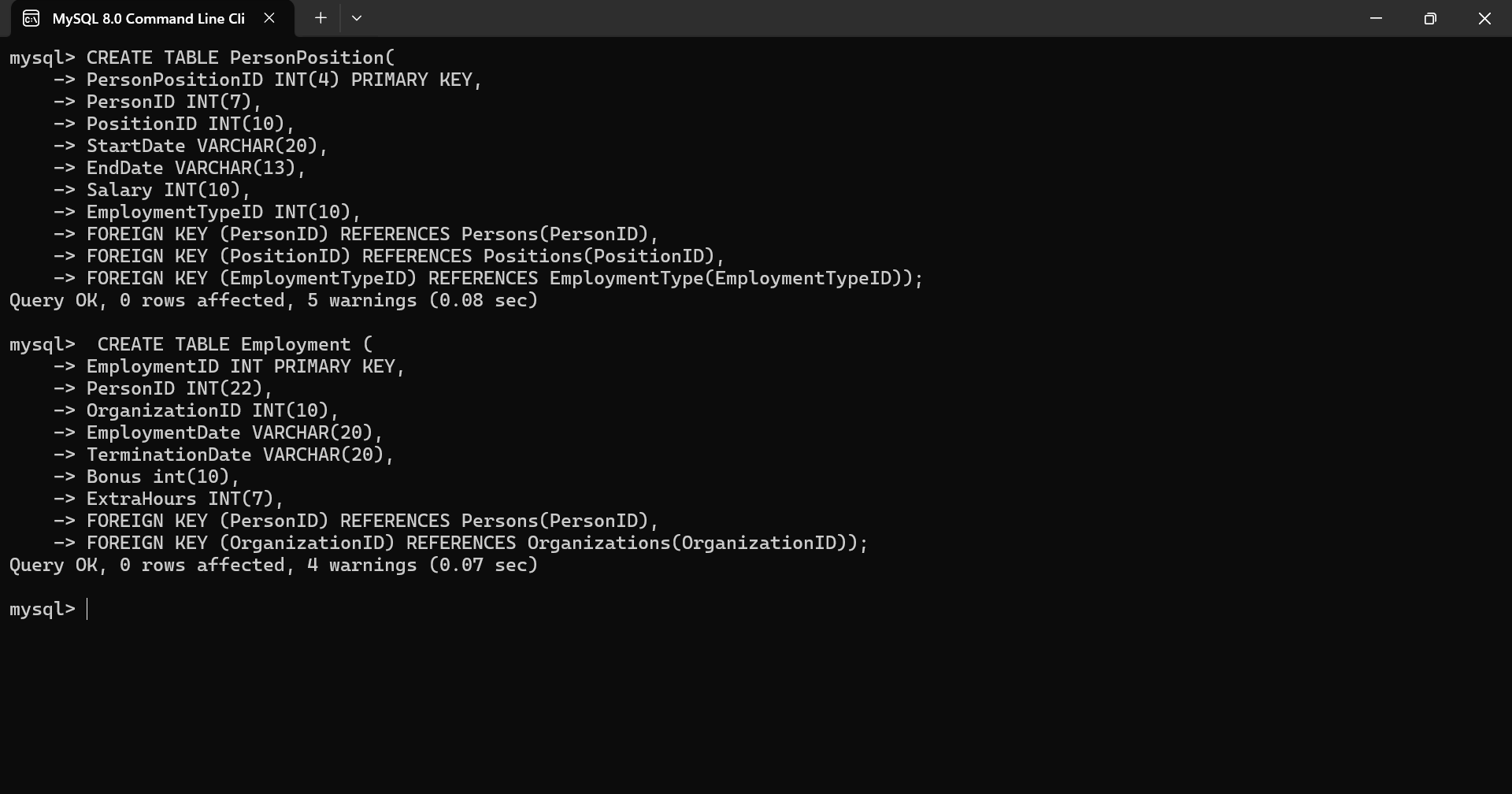


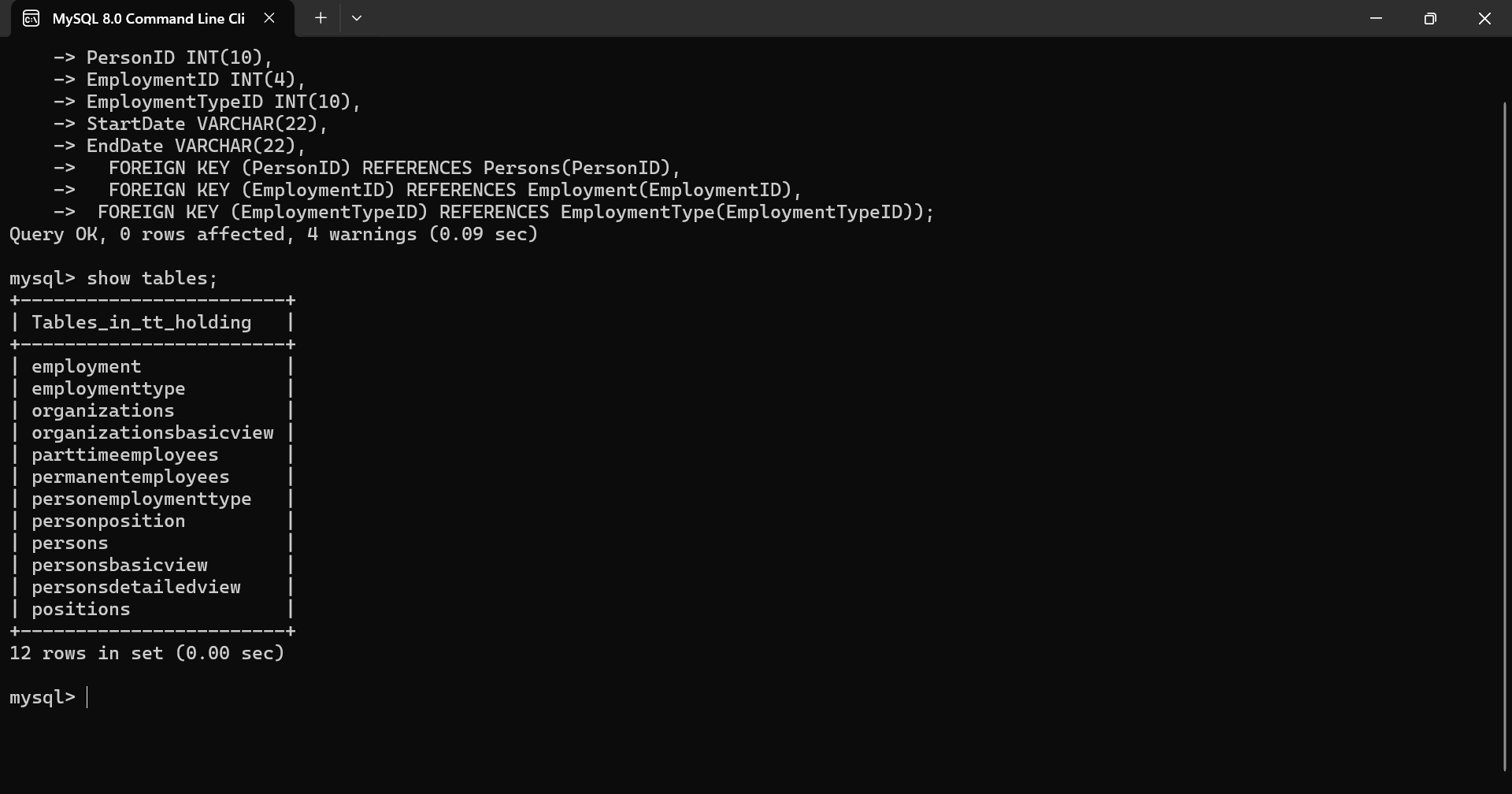


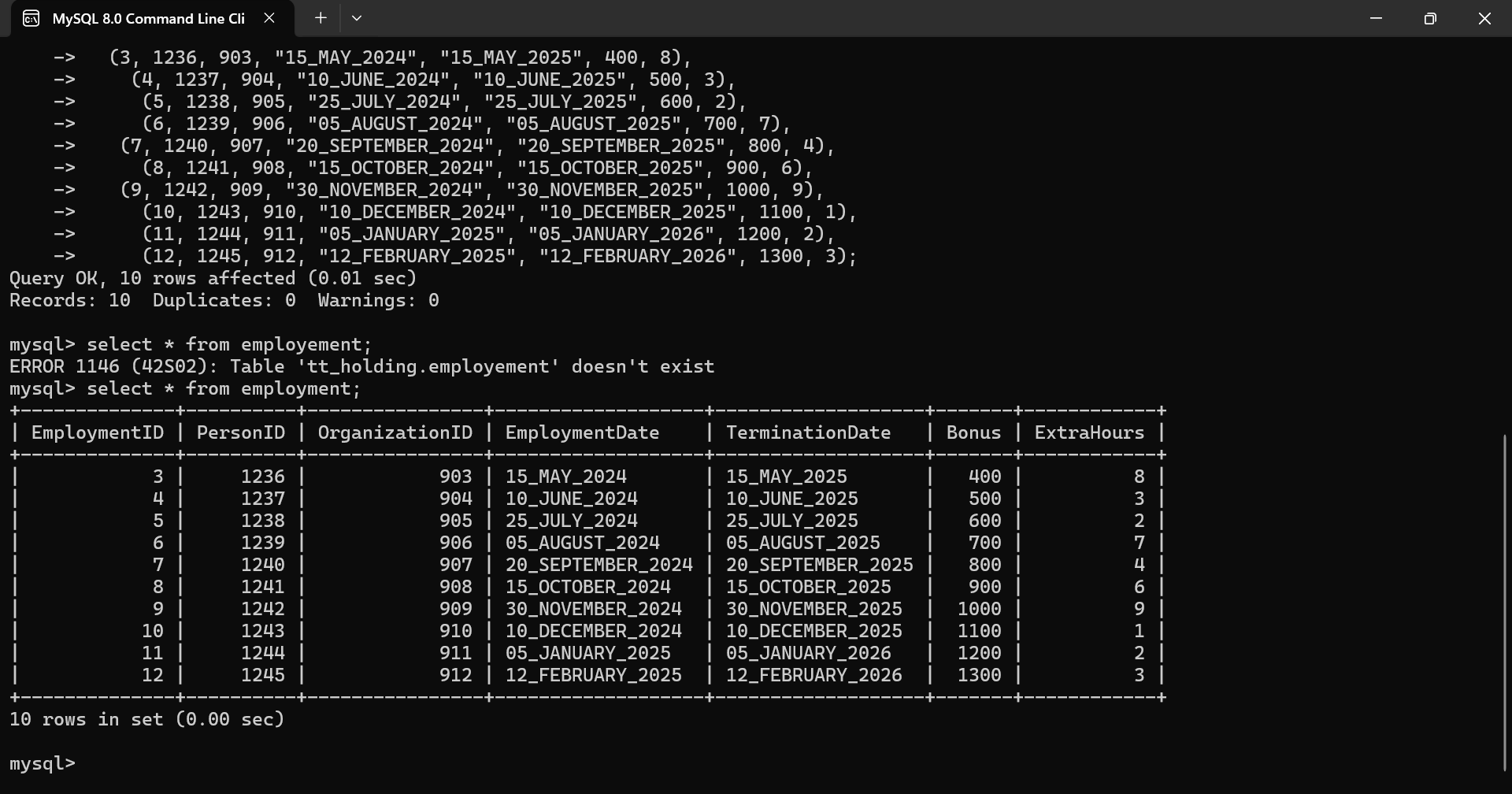
**Creating Views**

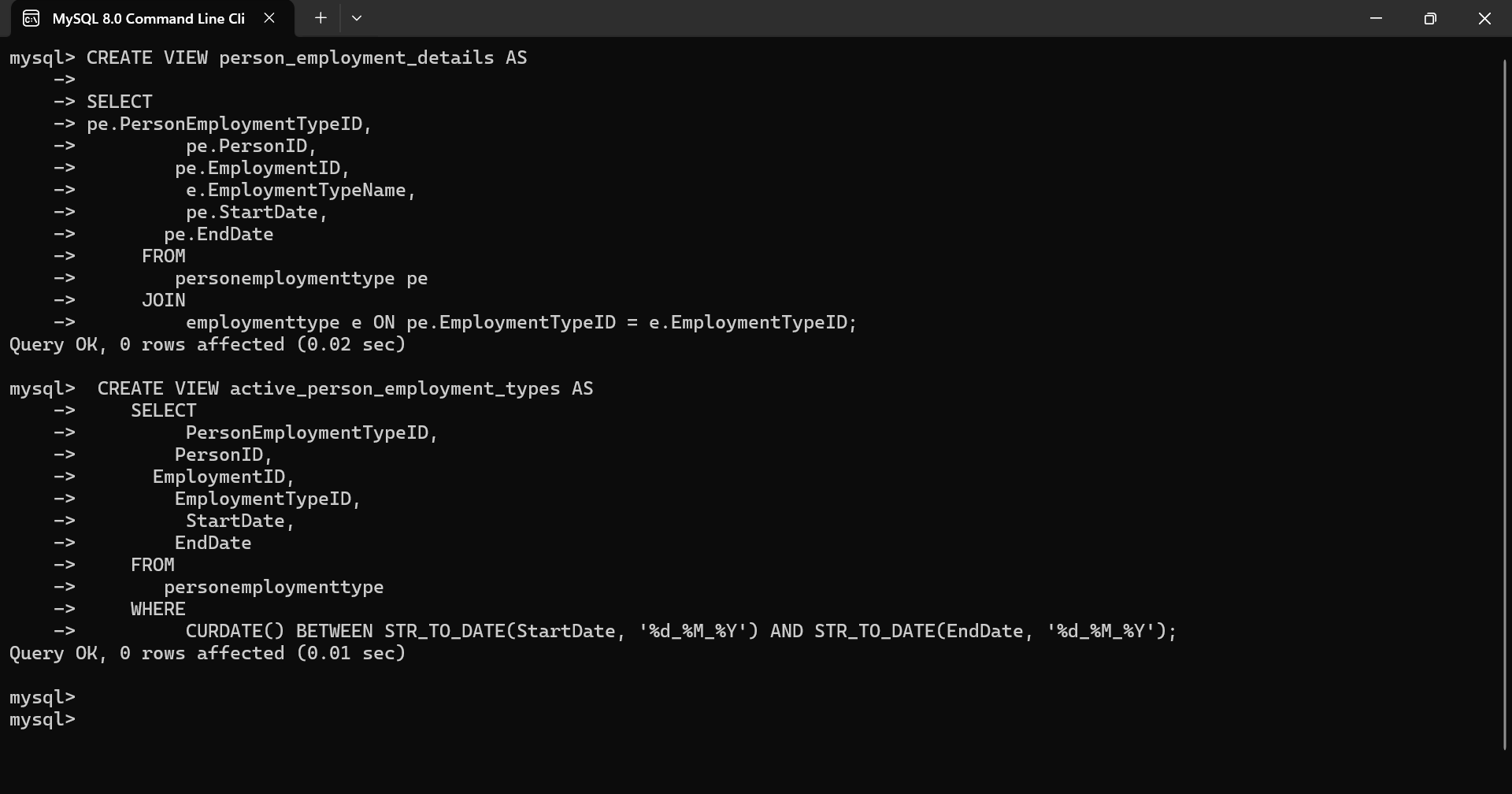
**View** is virtual table based on the result set of an sql statements.

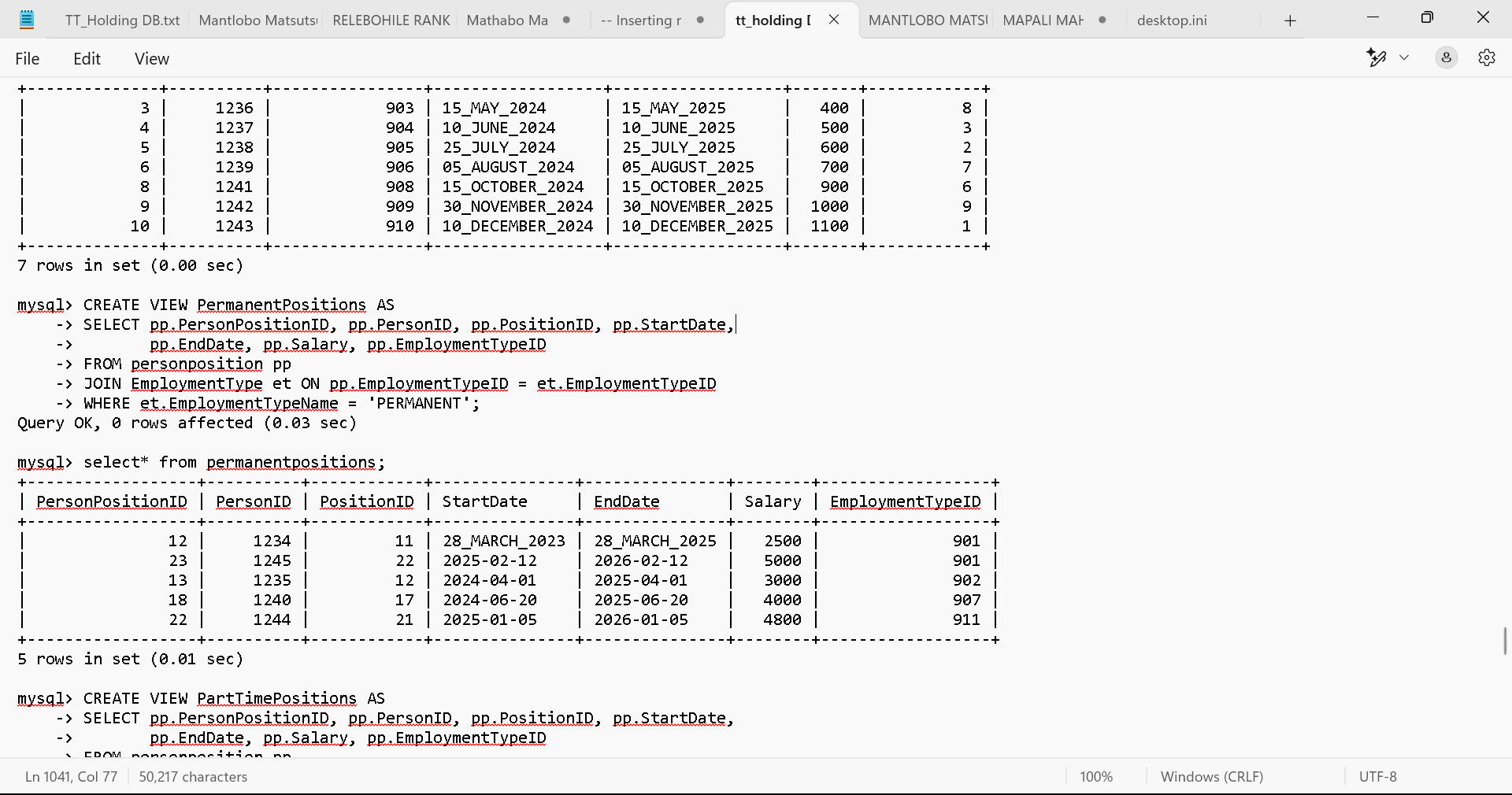






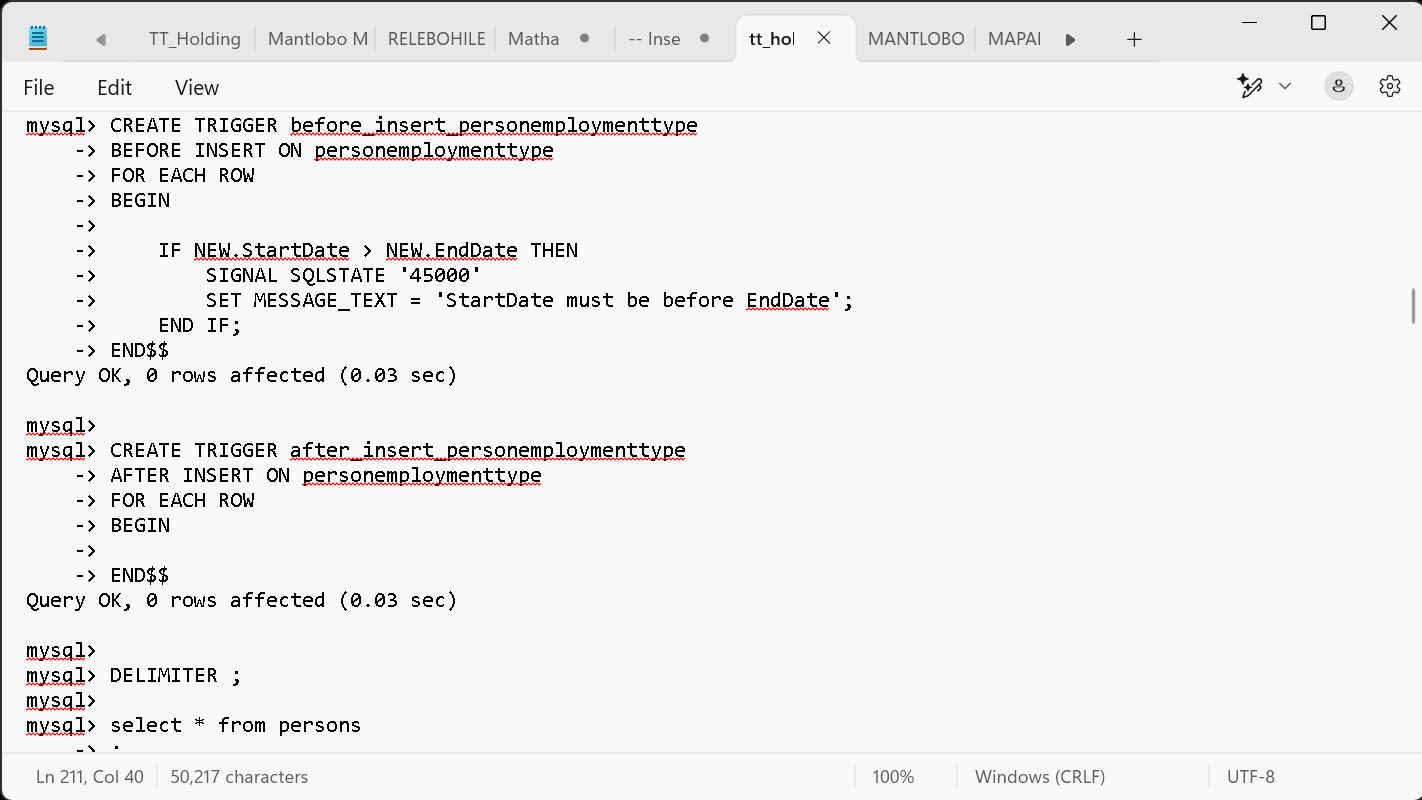


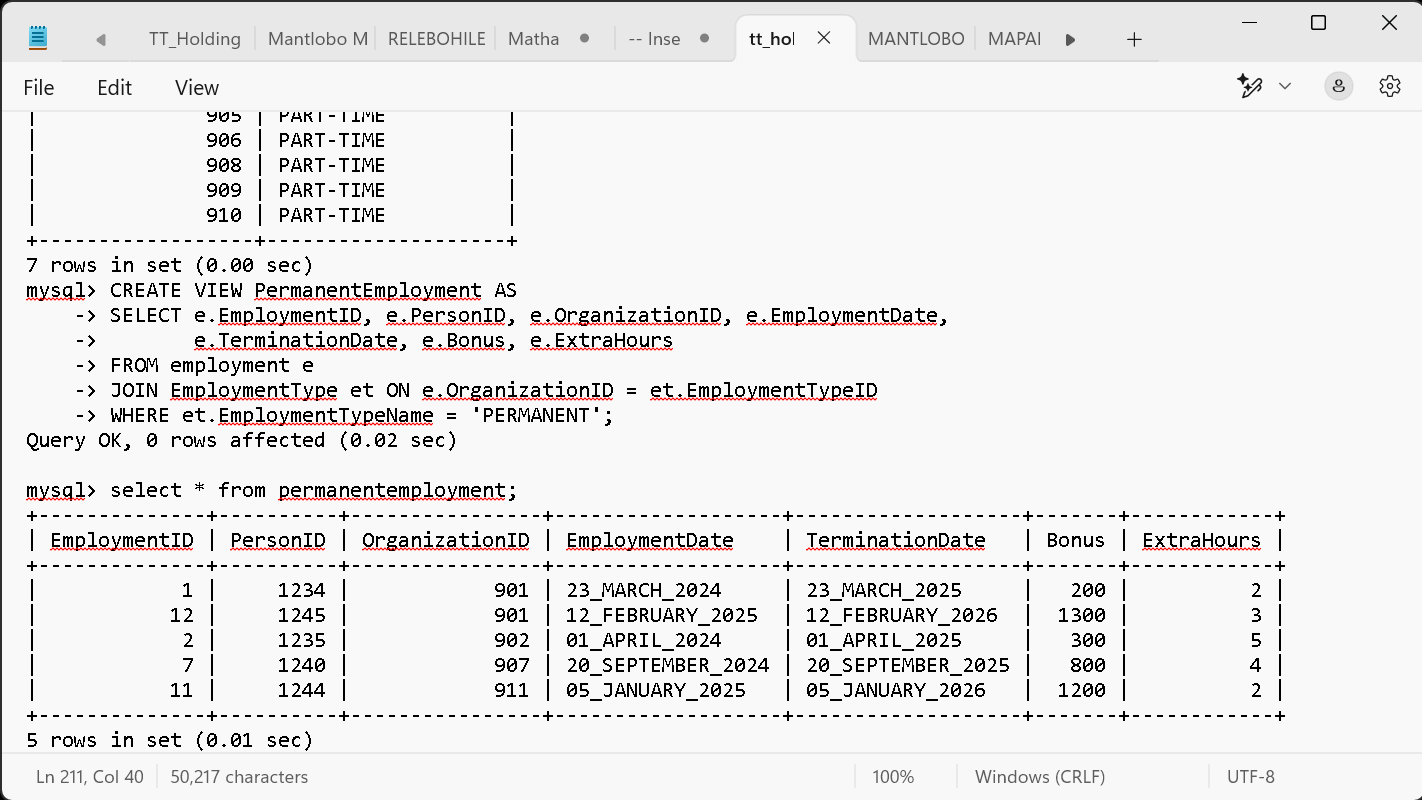


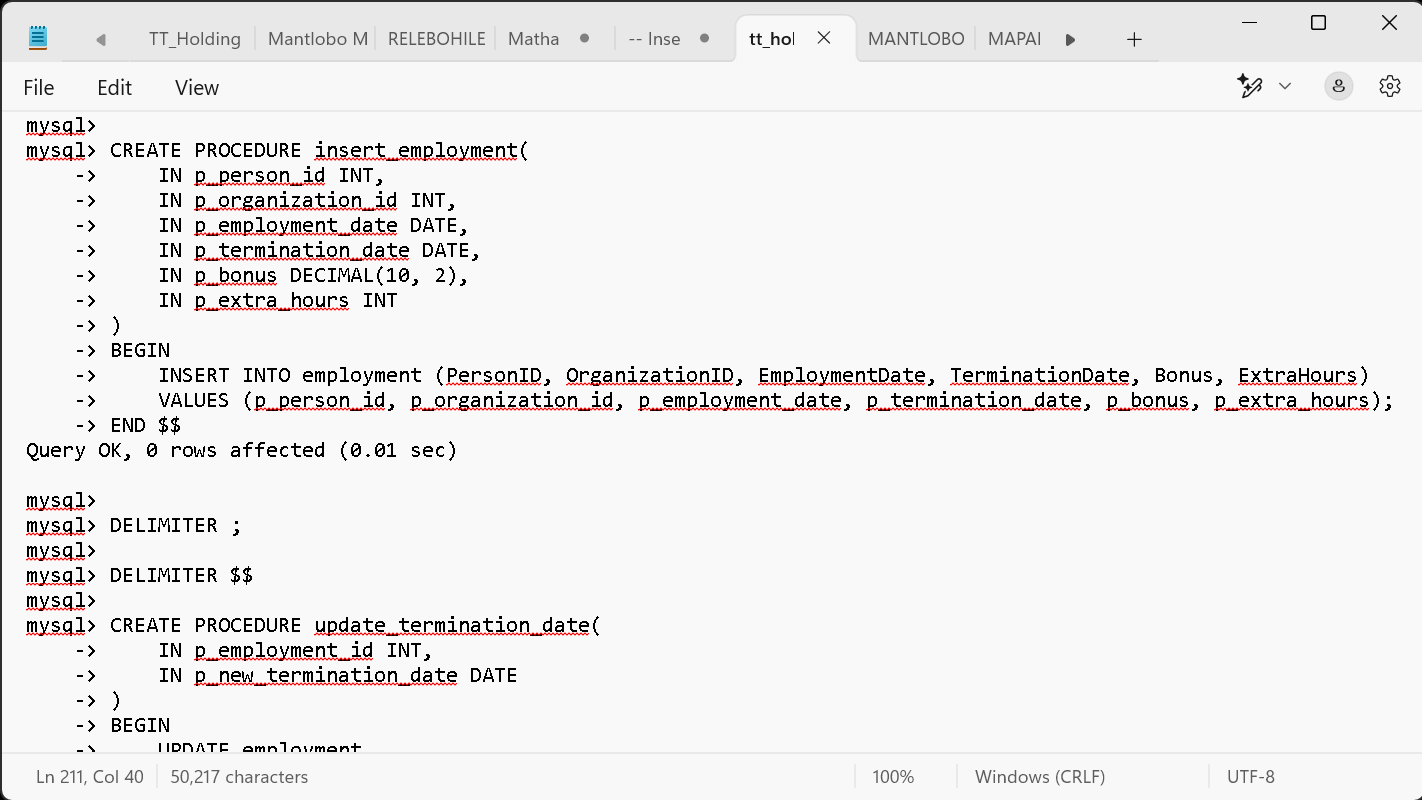


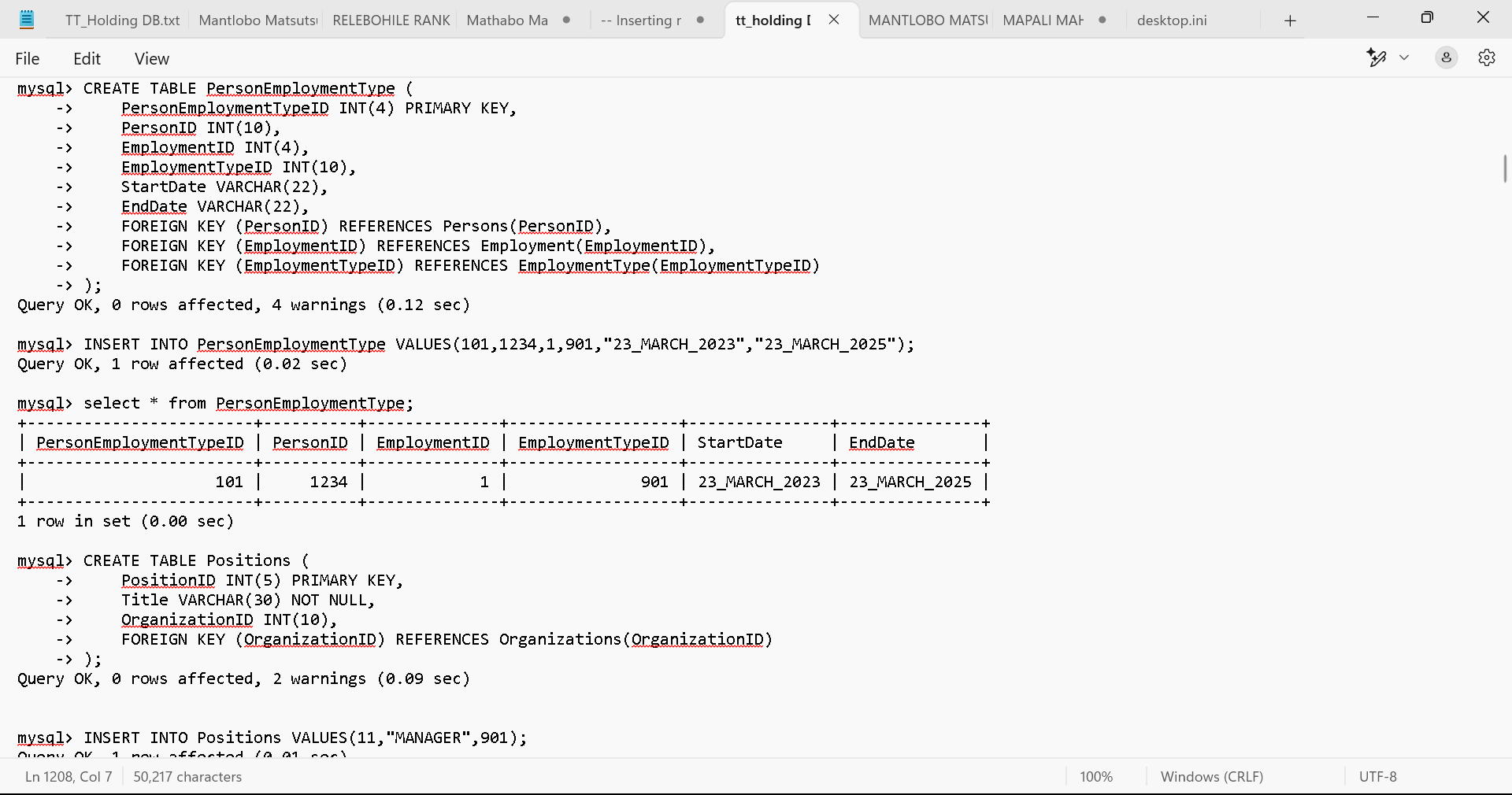
**Creating Triggers**

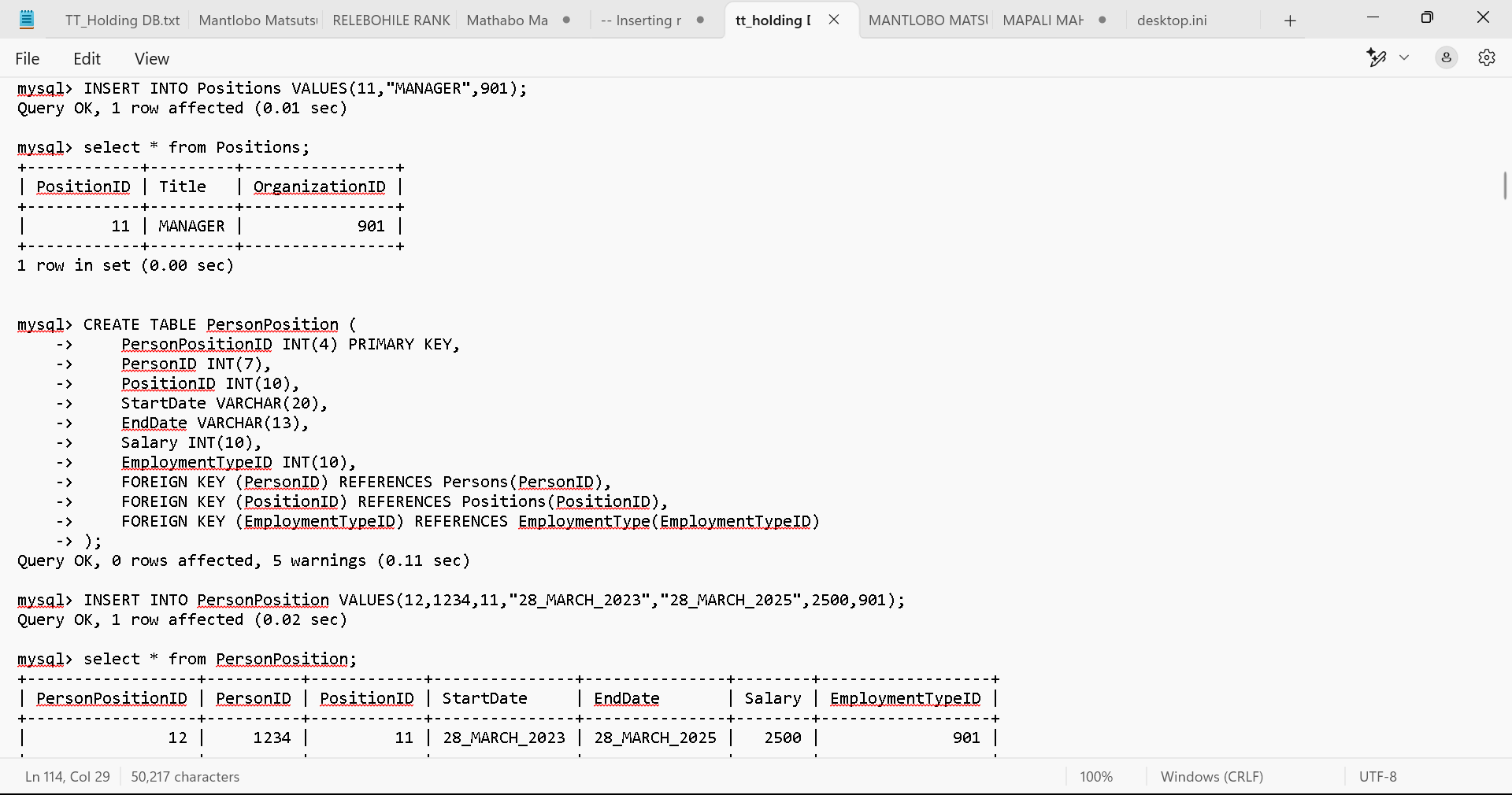
Triggers is a set of instructions automatically executing in response to a particular event.

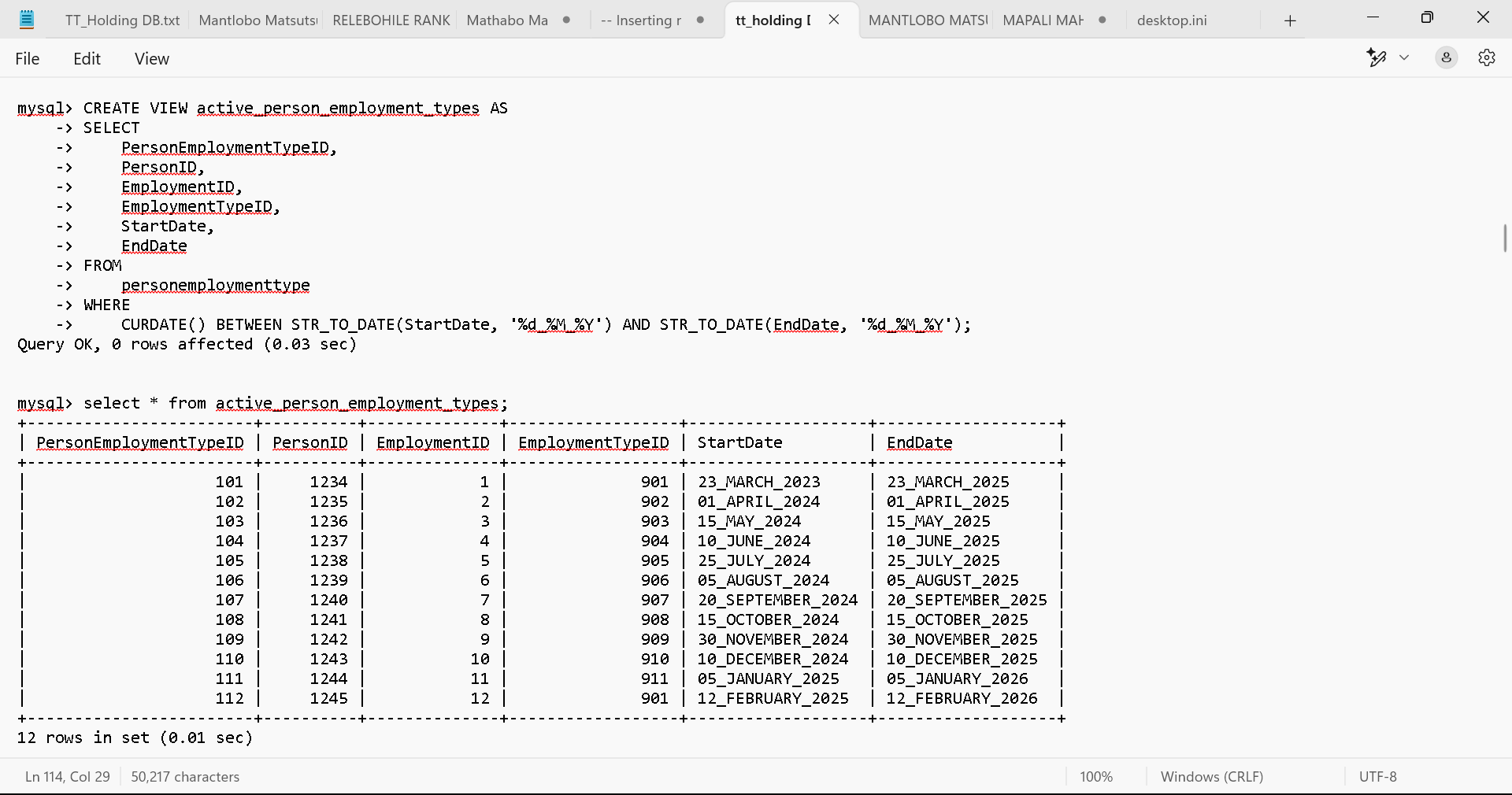


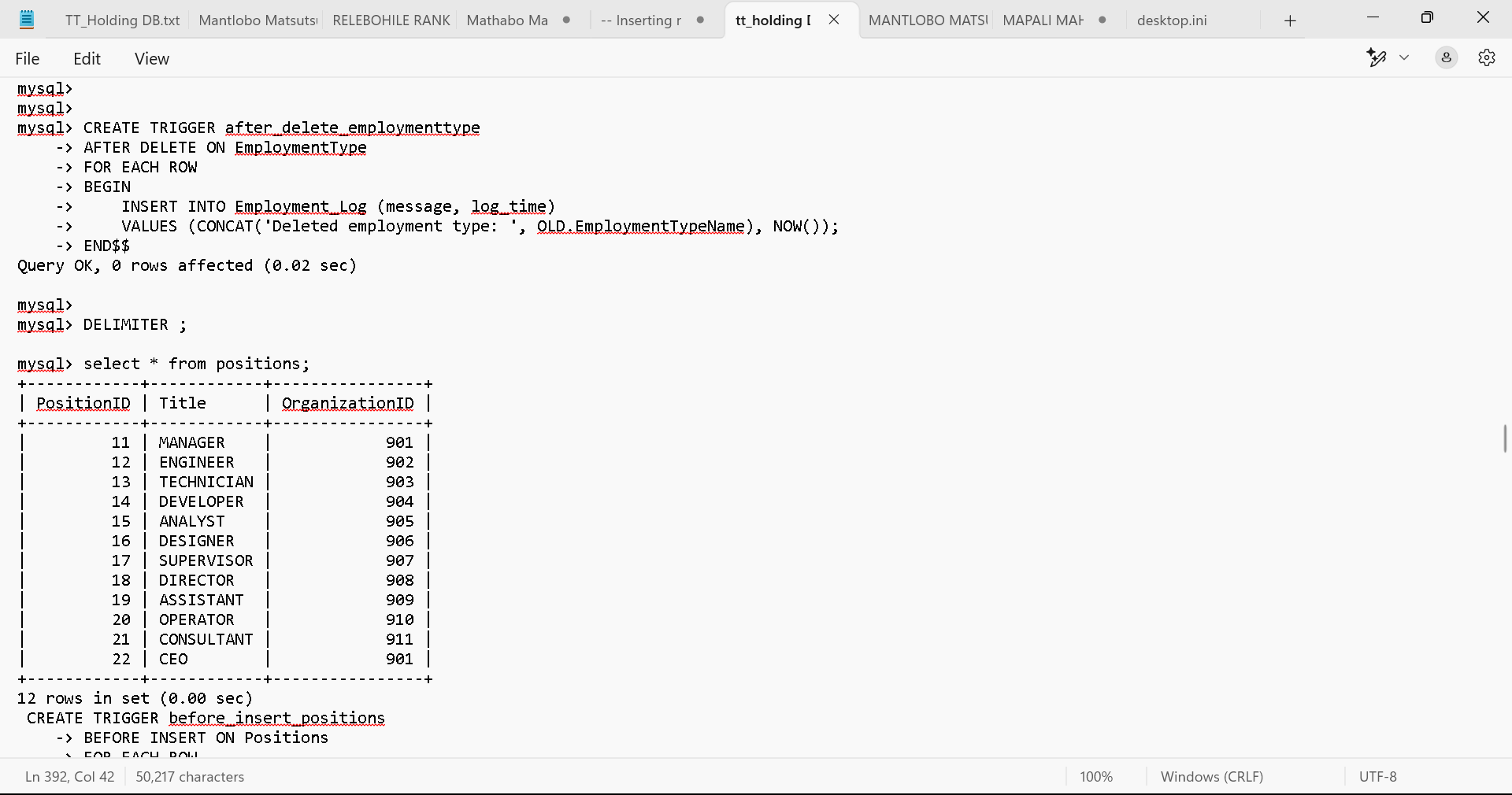








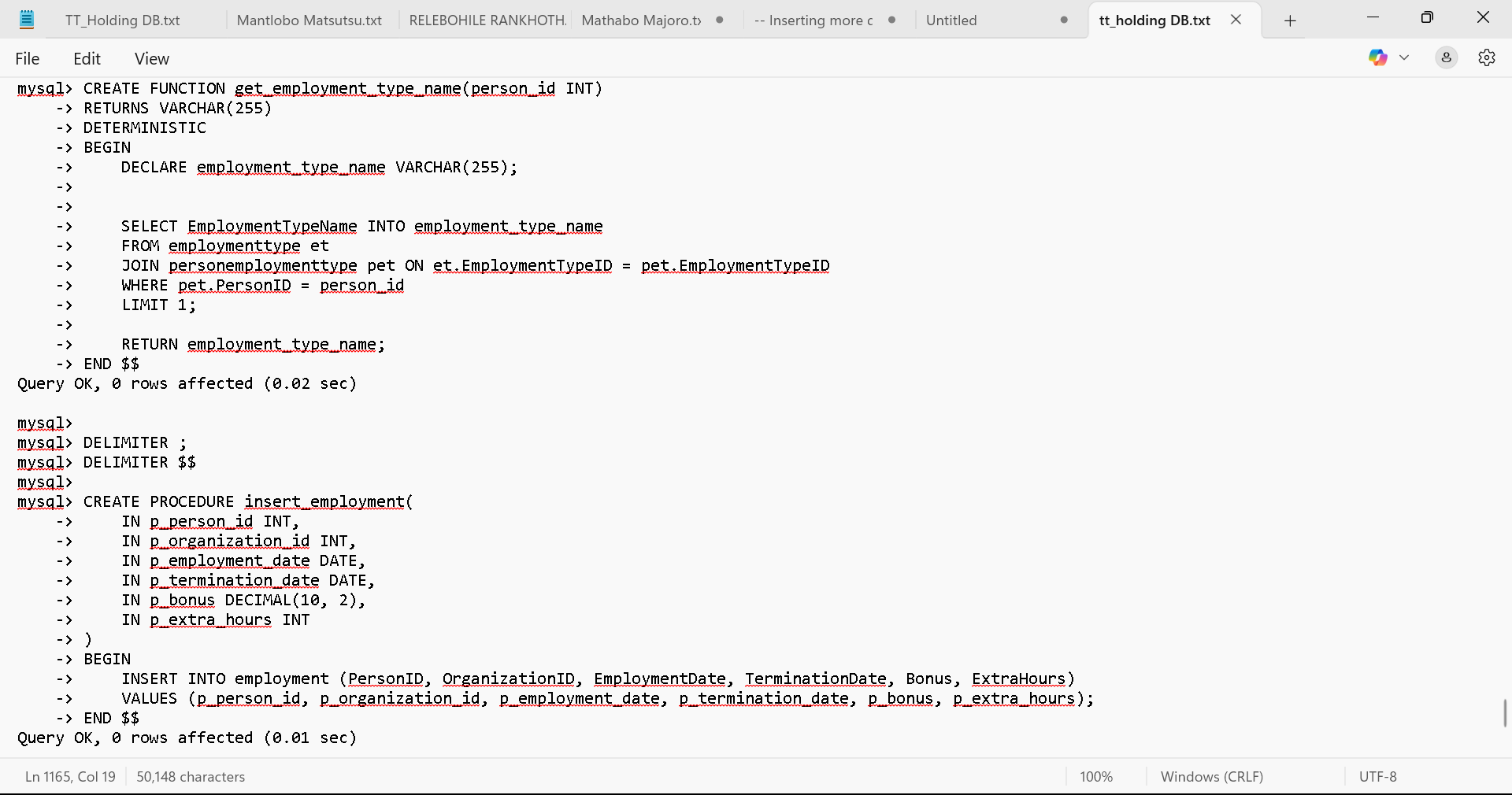




**Creating Functions and Procedures**

**Procedure** is a pre-written, reusable code block designed to perform a specific task or set of tasks.

**Functions** is pre-written blocks of code that perform specific tasks, taking input parameters, executing operations and returning a result**.**



**Creating User**

