**Table of Contents**

List of Figures …………………………………………………………………………………………………………………………………2

List of Tables …………………………………………………………………………………………………………………………………..3

List of Abbreviations ………………………………………………………………………………………………………………………4

Abstract ………………………………………………………………………………………………………………………………………….5

Introduction …………………………………………………………………………………………………………………………….6&7

Literature Review ………………………………………………………………………………………………………………………..8

Findings & Discussion …………………………………………………………………………………………………………………..9

Methodology …………………………………………………………………………………………………………………………10&17

Testing …………………………………………………………………………………………………………………………………………18

Conclusion ………………………………………………………………………………………………………………………………….19

References …………………………………………………………………………………………………………………………………..20

Appendices ………………………………………………………………………………………………………………………………….21

**LIST OF FIGURES**

Figure 1: Entity Relationship Diagram  
Figure 2: SQL Table Creation  
Figure 3: Data Insertion and Retrieval  
Figure 4: Creating a View  
Figure 5: Granting Privileges  
Figure 6: Triggers

**LIST OF TABLES**

Table 1: Primary and Foreign Key Relationships  
Table 2: Employee Data Sample  
Table 3: Example of SQL Triggers  
Table 4: User Role Privileges

**LIST OF ABBREVIATIONS**

* ERD: Entity-Relationship Diagram
* DBMS: Database Management System
* SQL: Structured Query Language
* HR: Human Resources
* PK: Primary Key
* FK: Foreign Key

**TT HOLDING DATABASE DESIGN AND IMPLEMENTATION**

**ABSTRACT**

It was intended to develop a database system to control the personnel and employees of the corporation TT Holding, to keep the personal information, the status of the employment (permanent or part time), job title, company with which employees are associated. To organize all this data we created an Entity-Relationship (ER) diagram linking employees, organizations, job positions, and job categories. A set of tables were developed using SQL scripts, it was extended (by appropriate data links) by SQL wizard. Advanced SQL features like triggers, stored procedures, views and so on were used to control and maintain the particular business rules and to keep the consistency of the data.

At some points during development there were problems that is triggers for update of pay, and deletion of employees. After a lot of testing and debugging, the issues were solved. For the highest security we assigned special access privileges to certain jobs like administrators and HR managers. The project ended here by providing a database system that automates some processes (checking employee data), saves some time and effort for admins and make daily activities very efficient. It is for sure that TT Holding will get a tool from which to go about it.

**INTRODUCTION**

**Problem Statement:**

TT Holding did now no longer have an incorporated gadget for coping with worker information, activity roles, and employment statuses. This ended in errors, inefficiencies in monitoring employees with numerous jobs in exclusive organizations, and problems preserving correct statistics of income changes and terminations. Without a robust database gadget, HR people and control struggled to keep worker records current, make well timed changes, and observe the identical requirements continuously during the organization.

**Problem Solving:**

TT Holding database design addresses these issues by developing automated systems that efficiently pursue employee information, positions, and roles within the company. With the help of the Hip (ER)diagram of Entity Relationships, the system clearly shows how employees, organizations, and roles are linked to ensure data consistency. The system also enforces restrictions to prevent reward reduction and ensure that people cannot be removed if the organization is still using triggers and saved processes.

**Objective:**

The primary objective of this project is to design and implement a comprehensive database system that can:

* Manage employee details, positions, and job statuses efficiently.
* Track how employees, organizations, and job positions are related.
* Keep data accurate using triggers and stored procedures.
* Make it easy to access and retrieve employee information.
* Automatically enforce rules, like stopping unauthorized changes to salary or job records.

**Scope & Constraints:**

A relational database that can take over employee information, recruitment, and roles within TT holdings is the focus of this project. By persecuting different work records for each employee, the system uses triggers and stored procedures to ensure that the data is correct and to make it easier for users to query the data in the view. Advanced features such as performance management, wages, pay, pay accounting are not available in the project. During the installation, there were several challenges with complex trigger issues such asemployee deletion and compensation adjustments, this was a problem.

**Database and ER Diagrams:**

**What is a database?**

A database is an electronically stored, systematic series of facts which could consist of words, numbers, images, videos, and different styles of files. Databases are controlled the use of specialised software program known as a Database Management System (DBMS), which permits customers to store, retrieve, and manage facts efficiently. Databases make sure facts consistency, integrity, and efficiency (GeeksforGeeks). In this project, the database will automate the control of worker facts, put into effect commercial enterprise regulations through triggers, and permit for clean querying and reporting.

**What is an ER diagram?**

An entity relationship diagram (ER diagram or ERD) is a visual representation of how items in a database relate to each other. ERDs are a specialized type of flowchart that conveys the relationship types between different entities within a system. They use a defined set of symbols, including rectangles, ovals and diamonds, and link them with connecting lines(Belcic and Stryker, 2024). ERDs provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization(TechTarget).

**LITERATURE REVIEW**

Current research and efforts on database design and employee management systems will be examined in this literature review. It will explore the use of databases in companies, the role of ER diagrams in system design and business rules used to ensure integrity of data. It will provide important insights into the architecture and functionality of the TT Holding database by taking into account past studies through this evaluation process.

**REVIEWING THE LITERATURE**

**1. Database Design in Employee Management Systems**

A well-structured database is important for managing employee data effectively. Modelers need to define the unique or primary key that references each element within a table. As a rule, every table always needs a primary key. And that primary key needs to be chosen with care. The relationships between tables are defined using foreign keys. A foreign key is a field within one table that refers to or is linked to another table’s primary key. This establishes the relationships between tables(Vertabelo,2023).

**2. Role of ER Diagrams in System Design**

An Entity-Relationship (ER) Diagram is a visual representation that illustrates the structure and relationships within a system. In the context of an Employee Management System, an ER Diagram depicts entities such as employees, departments, projects, and their relationships, such as an employee working in a department or being assigned to a project. The diagram outlines the key components, including entities (tables), attributes (fields), and the relationships between them (Creately, 2025). It shows how ER diagrams are important for showing relationships among entities such as employees, departments, payroll, projects and attendance. This helps make HR tasks easier and keeps data accurate.

3. **Utilization of Triggers, Stored Procedures, and Views**

The SQL Server storage procedure is a precompiled collection of SQL instructions and collections from flow instructions from optional control elements. They can be used to embody the logic of data manipulation, data validation, and business logic in SQL Server databases. These are many SQL instructions that can be saved and reused. A view in SQL is a virtual table based on the results of a SQL selection query. It does not store the data itself, but provides a way to access and manipulate data stored in one or more tables. Views can be used to restrict access to specific data (columns or rows) and provide data in a specific format, simplifying complex queries and improving security (Sharadika, 2024). A trigger is a stored procedure that is automatically executed when a particular action occurs. B. Enter, update, and delete data. They contribute to data consistency and can be used to ensure business rules are observed and to ensure that data is entered correctly.

**FINDINGS & DISCUSSION**

**Key findings:**

* ER diagrams improve data understanding
* Triggers automate data integrity
* Stored procedures and views help improve performance

These results are in line with what I hope the TT Holding database system will achieve: ER diagrams, when used effectively, can help to explain exactly how various bits of information related to each other can help to keep track of persons and their work in different business area. The use of triggers will automatically become conformant with business norms such as that there may be no unauthorized salary adjustments in making the data accurate and reliable. Stored procedures and views can make it easier and faster to get relevant information when necessary, taking care of the security of data such that it is safe and only authorized users can see the data.

**METHODOLOGY**

**Requirement Analysis**

**Functional requirements:**

1. **Employee Data Management** – Store and manage personal details of individuals in the **Person** table.
2. **Job Type Management** – The **Job Type** table should define different types of jobs available in the organization.
3. **Organization Structure Management** – The **Organization** table must store details of different departments or business units within TT Holding.
4. **Employment Records** – The **Employment** table must track employees' multiple roles within different organizations, ensuring accurate employment history.
5. **Position Management** – The **Position** table should define various job positions and their relationship with employees.
6. **Data Integrity and Relationships** – Use **primary keys** and **foreign keys** to establish strong relationships between tables and maintain accurate data.
7. **Business Rule Enforcement** – Triggers and stored procedures should ensure that salaries cannot be decreased and employees cannot be deleted if they are still working in the organization.
8. **User Access Control** – Restricts data access based on user roles such as HR, admin and employee.
9. **Efficient Querying and Reporting** – Views make it easier for HR to find and use employee data.

**Non functional requirements:**

1. Security – User authentication and authorization to protect employee data.
2. Scalability – The system should support future growth of the organization by allowing new job positions and departments.
3. Performance – The database should handle multiple employee records efficiently.
4. Reliability – The data should remain consistent through constraints and triggers.
5. Maintainability – The database should be easy to update and update as the business grows in future.

**System design**

The system is provided as a relational database at TT Holding in order to efficiently manage employee information, job roles and organizational positions. The system of archiving, retrieval and automatic controlling of rules is created by means of triggers and stored procedures. In order to support an efficient database management system the following table types are combined: Job Type, Organization, Employment, Person and Position the relationships and ties between all the data-types of the database are defined using Entity-Relationship (ER) modeling.

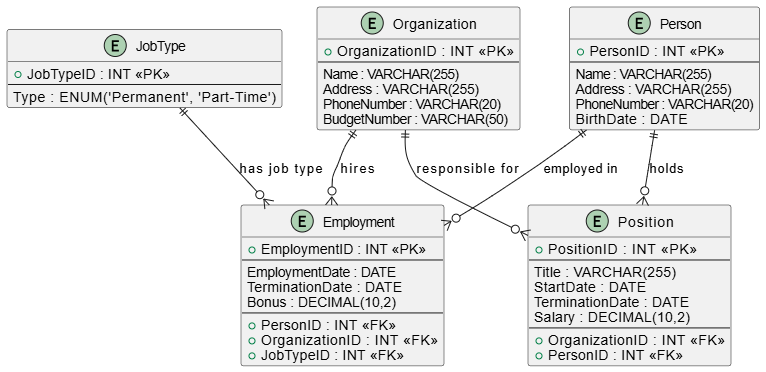
One can use this model to visualize the connections among various data types. The ER diagram provides a good overview of each individual's role in their organization and job role. To accomplish effective access to data one uses views to provide easy retrieval of information, while stored procedures and triggers automate important information pertaining to an individual's salary adjustments and the employment status.

**Architectural design**

Generally speaking the system has three layers: data, application, and user interface. Data LayerThe structure of the database is built according to a relational database framework, such that tables such as Job Type, Organization, Employment, Person, and Position store and maintain employee data related to specific employees. The application layer consists of triggers and stored procedures that ensure consistency and security of the data. Also the application layer links the database to the users' PC. The application layer then handles processing of data, validates the data and enforces business rules such as no decrease in salary and keeping employment records. The user interface layer also helps HR professionals and managers to access and view the employee data, access employee information, and effectively update records within the system. Therefore data flows smoothly, the system can be accessed easily, and employee information can be changed.

**ER Diagram**

Fig 1. Entity Relation Diagram



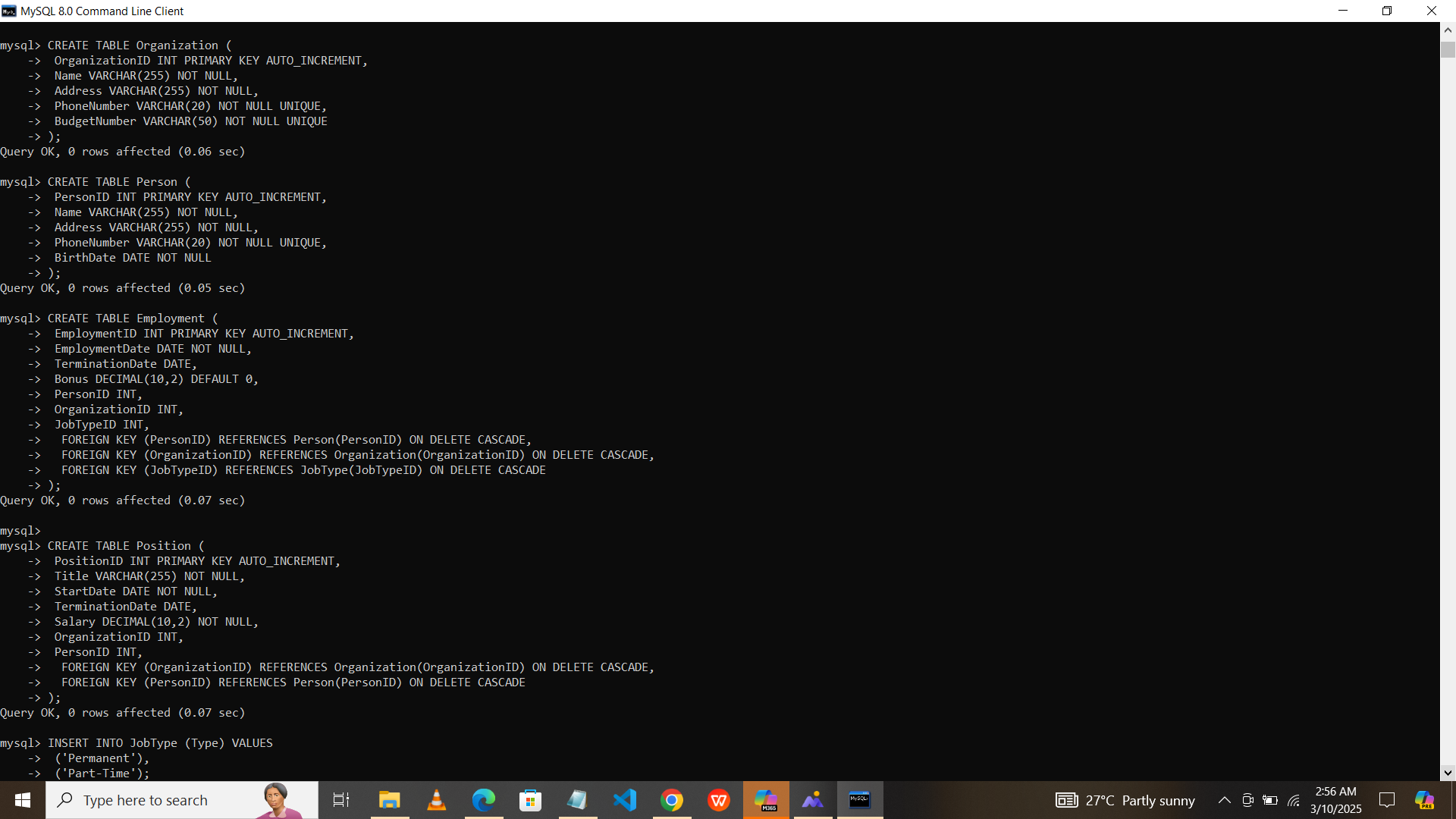
The ER diagram shows a clear structure of how data is organized, showing key companies such as people, Job type, positions, organizations, and employment documents. Relationships between entities ensure data consistency and eliminate redundancy.

**System Implementation / Prototyping**

The database was implemented using **SQL CREATE TABLE** statements, defining:

* **Primary Keys (PK)** for unique identification of records.
* **Foreign Keys (FK)** to maintain relationships between tables.
* **Constraints** such as NOT NULL, UNIQUE, and CHECK to enforce data accuracy.

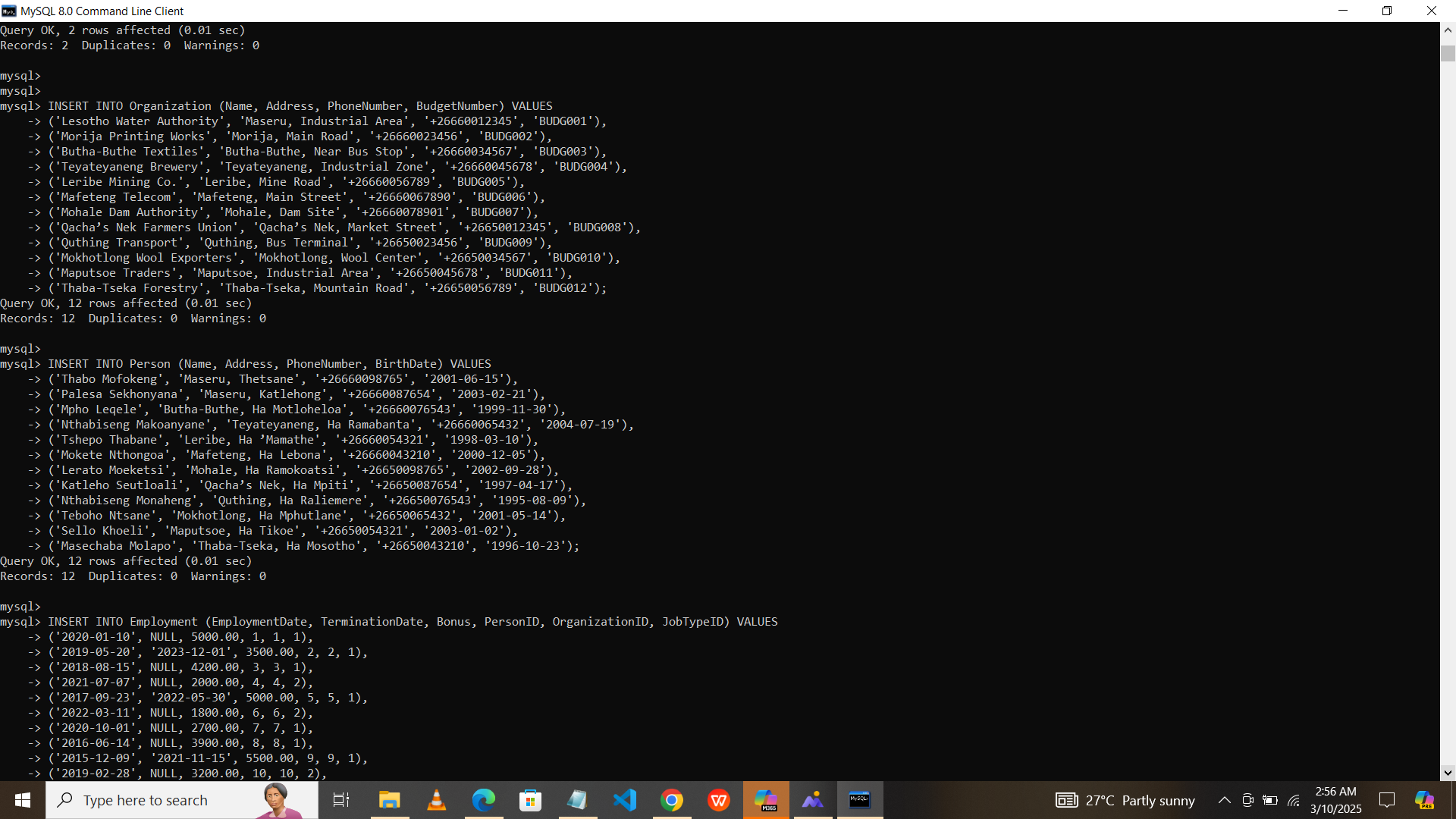
Fig 2. SQL table creation



**Data insertion and Retrieval**

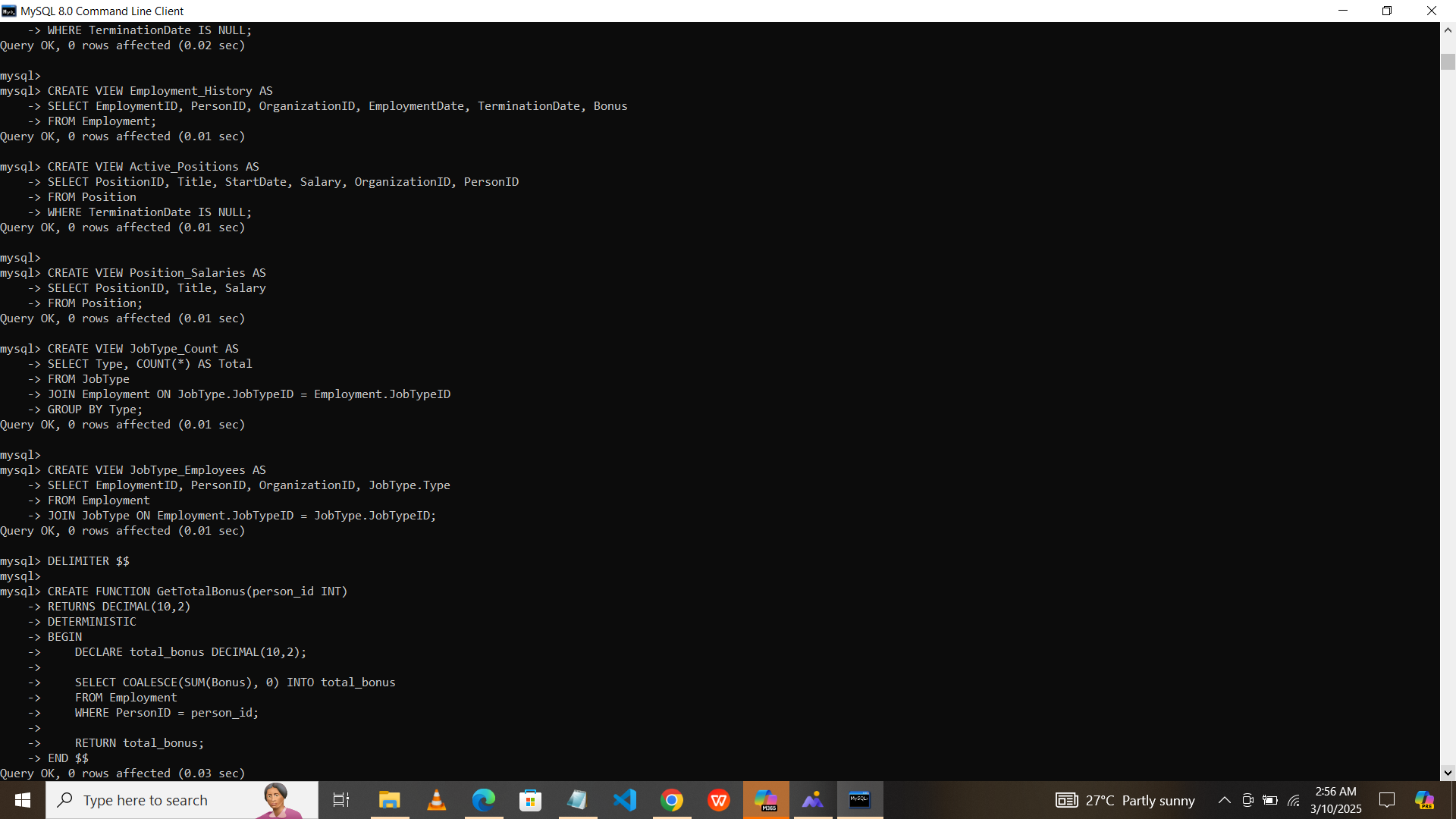
* **INSERT statements** were used to populate tables with sample data.
* **SELECT statements** were used to retrieve employee records efficiently.

Fig 3. Data insertion and Retrieval



**Creating a View**

Fig 4 Creating a View

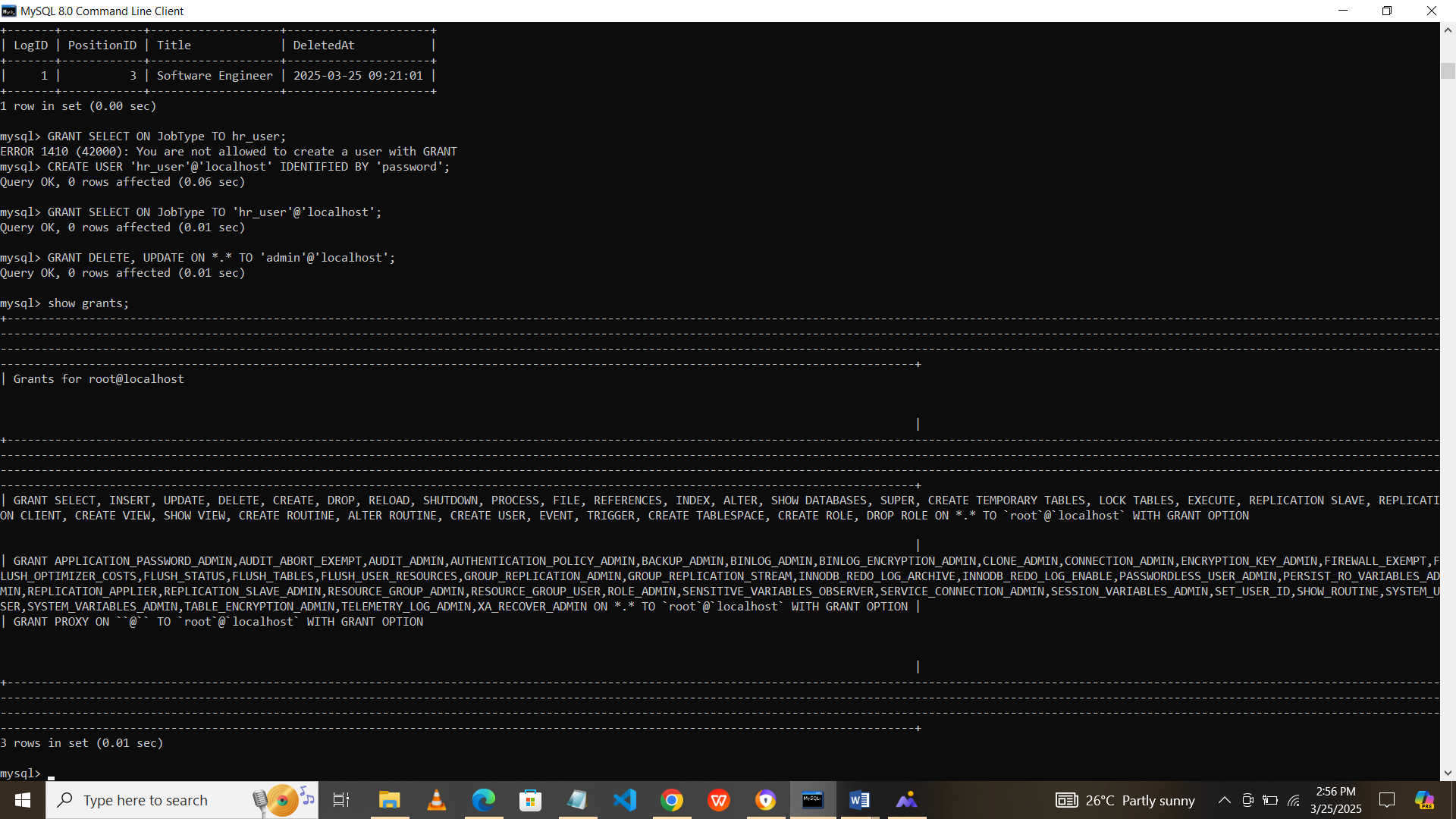


**Granting Privileges**

To maintain security, different users were assigned different access rights using SQL GRANT statements:

1. Admin: Can delete and update all the database tables.
2. HR manager: Selects but cannot update or delete employee records.
3. Employee: Can only view their records.

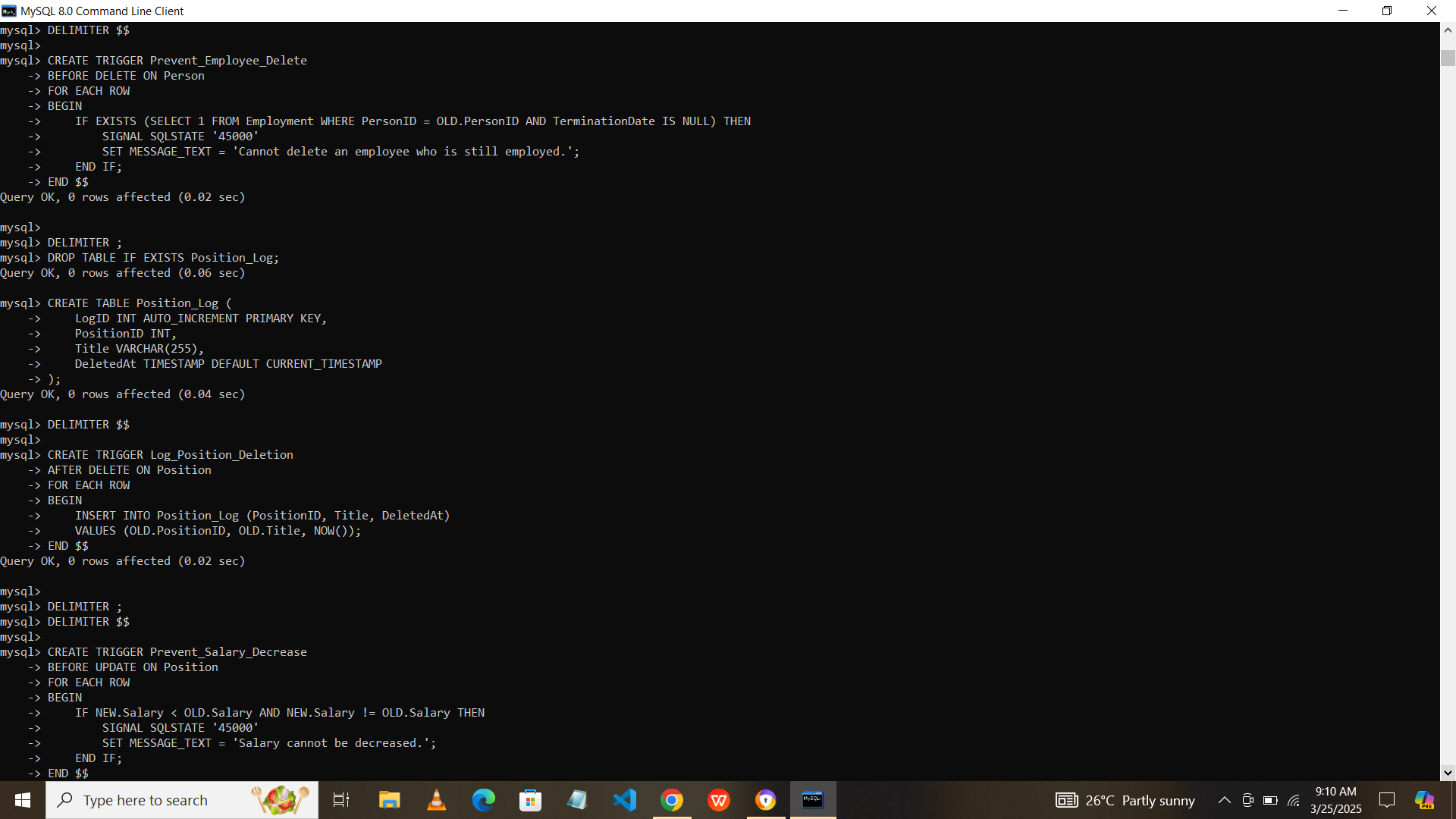
Fig 4 Granting Privileges



### ****Triggers for Business Logic Enforcement****

Triggers have been developed to implement key company guidelines. If the contract is in progress, you can also automate the process of determining termination data. These triggers reduce system human error and protect data consistency.

Fig 5 Triggers



### 

### TESTING

I examined the triggers to make certain they labored successfully and stored the information accurate. Here is what I did:

**1. Testing the Employment Trigger:**

* I brought a brand new document to the Employment table with the proper information. The document changed into brought successfully, and the Employment\_Log table recorded the time of the insertion, which confirmed that the cause labored as expected.
* The Position\_Log table did not have any entries, this means that that the cause did not do some thing for the Position table at that moment.

**2. Testing the Position Deletion Trigger:**

* + I attempted to delete a document from the Position table with PositionID = 3. The deletion went through, however no new access seemed within side the Position\_Log table. This intended that the cause for logging function deletions did not paintings as planned.
  + When I checked, I observed that the termination date was not set for the deleted function, which indicates that the cause intended to set termination dates was not activated either.

**3. Testing Salary Update Trigger:**

* + I brought a brand new function with a legitimate begin date and revenue to the Position table.
  + After deleting the equal function (PositionID = 3), the Position\_Log table successfully logged the deletion, together with the function`s identify and the time it changed into deleted, displaying that the cause for logging deletions labored properly.

Based on these tests, I confirmed that the deletion triggers worked for logging but not for establishing termination dates or performing other essential modifications. Otherwise, I will need to check the trigger logic to be certain that everything works properly.

**CONCLUSION**

In conclusion, TT Holding database system has solved the personnel data management, jobs job roles and organizational positions. The systems' design of relational database provides all the means for more effective staff knowledge management and organization. The system layout is developed on an Entity-Relationship (ER) diagram to ensure clearly displayed and maintained entity such as workers, jobs, organizations.

The use of advanced SQL tools such as triggers, stored procedures and views enabled the automation of very important business rules such as preventing unauthorized salary adjustments and ensuring correctness of employee records. There were some initial issues with triggers especially during employee deletions and salary adjustments but after extensive testing and debugging, they were fixed and the system was now dependable.

In addition, security was efficiently built to provide employees with separate access privileges based on their assigned work functions, to ensure confidential information is only available to authorized team members. With functions to facilitate quick search for data and automatic rule execution, the system significantly streamlines HR procedures, reduces errors, and ensures appropriate data management. The TT Holding database system is a flexible, reliable, and secure platform for data management in personnel information systems and job functions for HR purposes.

**REFERENCES:**

1. Creately (2025) '6 ER Diagrams for Employee Management System + Free Templates' [Online]. Available at: <https://creately.com/guides/er-diagram-for-employee-management-system/>. (Accessed: 25 March 2025).
2. GeeksforGeeks (2025) 'What is Database?' [Online]. Available at: <https://www.geeksforgeeks.org/what-is-database/>. (Accessed: 25 March 2025).
3. Madu Sharadika (2024) 'Stored Procedures, Views, Triggers and Cursors in SQL.' Medium [Online]. Available at: <https://medium.com/%40madu.sharadika/stored-procedures-views-and-triggers-in-sql/>. (Accessed: 25 March 2025).
4. Ivan Belcic and Cole Stryker (2024) 'What is an Entity Relationship Diagram?' IBM [Online]. Available at: <https://www.ibm.com/think/topics/entity-relationship-diagram/>. (Accessed: 25 March 2025).
5. Vertabelo (2023) 'How Do You Design an ER Diagram for an Employee Database?' [Online]. Available at: <https://vertabelo.com/blog/er-diagram-for-employee-database>. / (Accessed: 25 March 2025).

**APPENDICES:**

**Appendix A:** Sample SQL Code for Table Creation

**Appendix B:** Sample Data Entries

**Appendix C:** Example SQL Query for Data Retrieval

**Appendix D:** Trigger Testing Logs

**Appendix E:** User Role and Privilege Definitions

