# Title of the System

TT Holding Comprehensive Employment Management System

# Table of Contents

[1 Title of the System 1](#_Toc193931495)

[2 Table of Contents 2](#_Toc193931496)

[3 List of Figures 4](#_Toc193931497)

[4 List of Tables 4](#_Toc193931498)

[5 List of Abbreviations 5](#_Toc193931499)

[6 Abstract 6](#_Toc193931500)

[7 Chapter 1 : Introduction 7](#_Toc193931501)

[7.1 Problem Statement 7](#_Toc193931502)

[7.2 Problem Solving 7](#_Toc193931503)

[7.3 Objective 7](#_Toc193931504)

[7.4 Scope & Constraints 7](#_Toc193931505)

[8 Chapter 2 : Literature Review 7](#_Toc193931506)

[8.1 Introduction 7](#_Toc193931507)

[8.2 Reviewing the literature 7](#_Toc193931508)

[8.2.1 Concept of relational model 7](#_Toc193931509)

[8.2.2 Advanced SQL 8](#_Toc193931510)

[8.3 Findings and Discussions 8](#_Toc193931511)

[9 Chapter 3 : Methodology 8](#_Toc193931512)

[9.1 Requirement Analysis 8](#_Toc193931513)

[9.2 System Design 8](#_Toc193931514)

[9.2.1 Architectural Design 8](#_Toc193931515)

[9.2.2 ER Diagram 9](#_Toc193931516)

[9.3 System Implementation/Prototyping 9](#_Toc193931517)

[9.4 Testing 17](#_Toc193931518)

[9.4.1 Data Insertion Testing 17](#_Toc193931519)

[10 Chapter 4 : System Initiation and Planning 17](#_Toc193931520)

[10.1 Assessing Project Feasibility 17](#_Toc193931521)

[10.2 Project Plan 18](#_Toc193931522)

[11 Chapter 5 : System Analysis 19](#_Toc193931523)

[11.1 Determining System Requirements 19](#_Toc193931524)

[11.2 Structuring System Requirements 19](#_Toc193931525)

[12 Chapter 6 : Conclusion 20](#_Toc193931526)

[12.1 Advantages of the System 20](#_Toc193931527)

[12.2 Future Enhancement of the System 21](#_Toc193931528)

[12.3 Potential Benefit 21](#_Toc193931529)

[12.4 Conclusion 21](#_Toc193931530)

[13 References 22](#_Toc193931531)

[14 Appendices 23](#_Toc193931532)

[14.1 Appendix A : SQL code 23](#_Toc193931533)

[14.1.1 Table creation 23](#_Toc193931534)

[14.1.2 Triggers 24](#_Toc193931535)

[14.2 Appendix B: User privileges 25](#_Toc193931536)

# List of Figures

|  |  |
| --- | --- |
| Figure no. | Figure Name |
| Figure 1 | ER diagram |
| Figure 2 | Screenshot: Creating Person, Organization, Employment, Jobposition Tables |
| Figure 3 | Screenshot: Creating PositionHistory, permanentemployee & Temporaryemployee tables |
| Figure 4 | Screenshot: Inserting data into Person & Organization tables |
| Figure 5 | Screenshot: Inserting data into Employment & Jobposition tables |
| Figure 6 | Screenshot: Inserting data into PermanentEmployee & TemporaryEmployee tables |
| Figure 7 | Screenshot: Inserting data into PositionHistory table |
| Figure 8 | Screenshot: Creating views for Person, Organization & Employment tables |
| Figure 9 | Screenshot: Creating views for Jobposition, PermanentEmployee, TemporaryEmployment tables |
| Figure 10 | Screenshot: Creating views for PositionHistory table |
| Figure 11 | Screenshot: Creating triggers |
| Figure 12 | Screenshot: Creating triggers |
| Figure 13 | Screenshot: Creating triggers |
| Figure 14 | Screenshot: Creating Functions |
| Figure 15 | Screenshot: Storing Procedures |
| Figure 16 | Screenshot: Testing wrong data insertion |

# List of Tables

|  |  |  |
| --- | --- | --- |
| Table No. | Table Name | Description |
| Table 1 | Person | Stores information of Employees |
| Table 2 | Organization | Stores company details |
| Table 3 | Employment | Tracks employment relationships |
| Table 4 | JobPosition | Stores positions the employee holds |
| Table 5 | PositionHistory | Tracks position changes |
| Table 6 | PermanentEmployee | Stores information for permanent employees |
| Table 7 | TemporaryEmployee | Stores information for temporary employees |

# List of Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Full Form |
| ER | Entity Relationship |
| SQL | Structured Query Language |
|  |  |

# Abstract

The TT Holding Comprehensive Employment Management System was created to manage organizational and personnel data efficiently. Using a relational database approach, this solution makes it easier to store, retrieve, and manage employment histories, job positions, pay, and organizational structures. Tables devoted to workers, organizations, employment relations, and job positions are essential database components.

Advanced SQL capabilities like views, triggers, stored procedures, and functions are used to improve data integrity, enforce business rules, and automate repetitive database processes. System security is maintained by privilege management and user access controls.

Numerous benefits are offered by the system, such as better decision-making through structured data structures, decreased administrative responsibilities, and higher efficiency in managing employment records. A web-based interface, sophisticated reporting features, and payroll system integration are possible future additions. The design, implementation, and possible improvements of the database are thoroughly examined in this report in order to provide TT Holdings with an employment management system that is optimum.

# Chapter 1 : Introduction

## Problem Statement

TT Holding company requires a system to manage both permanent and temporary employees. The company needs a system to efficiently track employee data, employment history, positions held, and organizational units.

## Problem Solving

This project is about building a system to organize and manage important information. The system will support data integrity, relationships between entities, and advanced SQL features such as views, triggers, and functions. These features ensure that the database supports efficient data management.

## Objective

The primary objectives of this project are to:

* Design a comprehensive database schema to capture all necessary data.
* Implement the database using MySQL.
* Populate the database with sample data.
* Utilize advanced SQL features to ensure data integrity and automate tasks.
* Create views for efficient data retrieval.
* Implement triggers to enforce business rules and maintain data consistency.
* Develop functions for data calculations and transformations.
* Create stored procedures for complex data operations.
* Grant appropriate privileges to normal users.

## Scope & Constraints

The project scope includes database design, table creation, and advanced SQL operations. The constraints are: limited 10-20 rows of data per table, advanced SQL features, creating two normal users with specific privileges.

# Chapter 2 : Literature Review

## Introduction

The purpose of the literature review is to find best practices and provide guidance for the database system design decisions made by TT Holdings. This literature review explores relevant database management principles, advanced SQL functionalities, and best practices in relational database design.

## Reviewing the literature

### Concept of relational model

(Chen, 2023) stated that relational model consists of tables, known as relations, which organize data in a structured way using rows and columns. Each row represents a unique record, and each column indicates a specific characteristic of that record.

### Advanced SQL

According to (Mullins, 2012), SQL views, a part of physical database design, are virtual "tables" created from SELECT statements to support specific application needs. While not necessary for database access, views offer a logical representation of data stored in underlying tables or other views, without requiring their own physical storage.

The use of triggers, stored procedures, and views enhances the functionality of employment databases. Triggers can be used for automating employment status updates, while stored procedures optimize payroll processing (Gupta, P., Agarwal, R. & Verma, A, 2016). Views play a crucial role in restricting data access while improving query performance (Huang, L., Park, J. & Kim, D, 2018). Implementing these techniques ensures that employment records remain accurate and secure.

## Findings and Discussions

A number of important conclusions pertinent to the TT Holdings database system are highlighted in the examined literature:

* Monitoring employment history is crucial. A thorough work history for every employee is ensured by using a distinct PositionHistory table.
* Efficiency is increased by using relational models. According to the studies, payroll processing can be optimized by dividing permanent and temporary workers.
* Functionality is improved by SQL automation; triggers, views, and stored procedures are used to help ensure data consistency.

# Chapter 3 : Methodology

## Requirement Analysis

The first step in methodology involved a thorough examination of TT Holding database design which included identifying the key entities such as Persons, Employment, Organization, Position, PositionHistory, PermanentEmployee, TemporaryEmployee with their associated attributes. The relationships between entities were carefully identified. The system manages both permanent and part-time employees and tracks salary changes, employee terminations, and position changes.

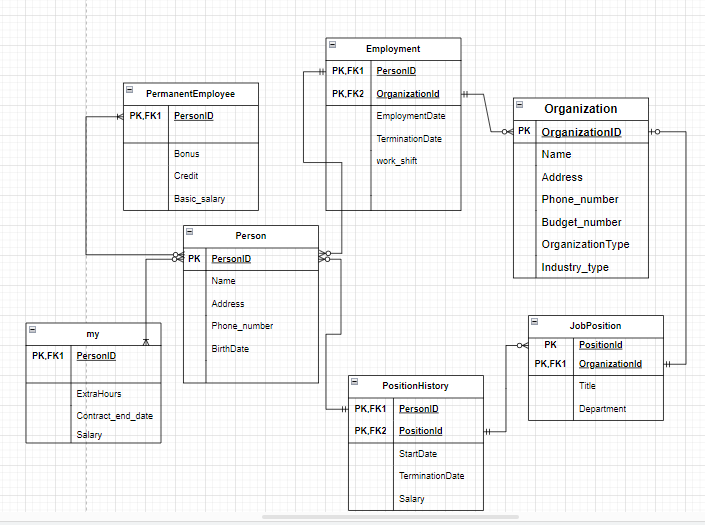
## System Design

### Architectural Design

Seven interconnected tables make up the relational database structure of the TT Holdings database system. The Person table, which is linked to Employment, PermanentEmployee, and TemporaryEmployee, is at the center. The Organization table, which is related to Employment and Position, includes data about several organizational units. The Employment table serves as the basis for maintaining PositionHistory and acts as a junction table, addressing the many-to-many interaction between Person and Organization. Job positions are described in the Position table, and the assignments of people to these roles throughout time are recorded in PositionHistory. Furthermore, the Person object is enhanced with particular attributes pertinent to each type of employment via the PermanentEmployee and TemporaryEmployee tables.

### ER Diagram

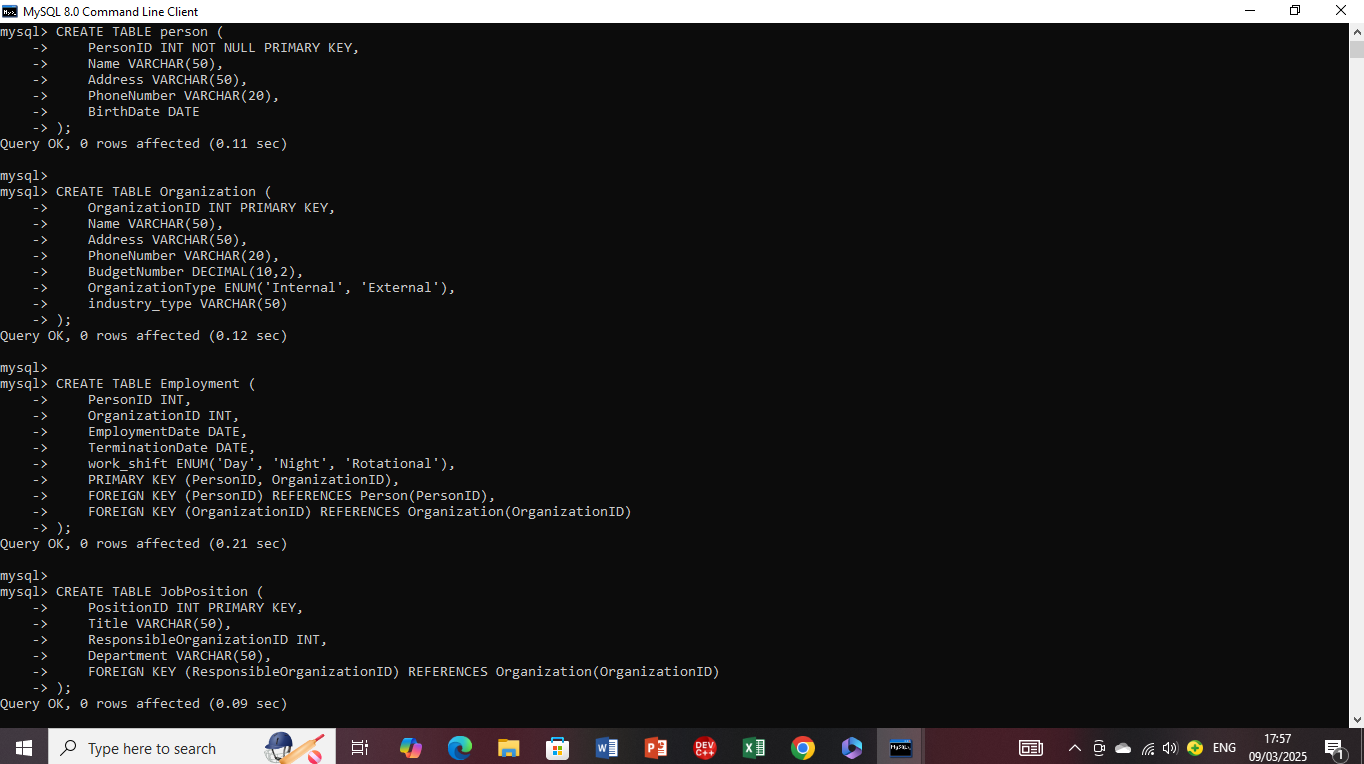
The design of the TT Holdings database is visually represented by the ER diagram. This diagram shows the seven key entities and the relationships between them, including their cardinalities and the primary and foreign key constraints that define these relationships



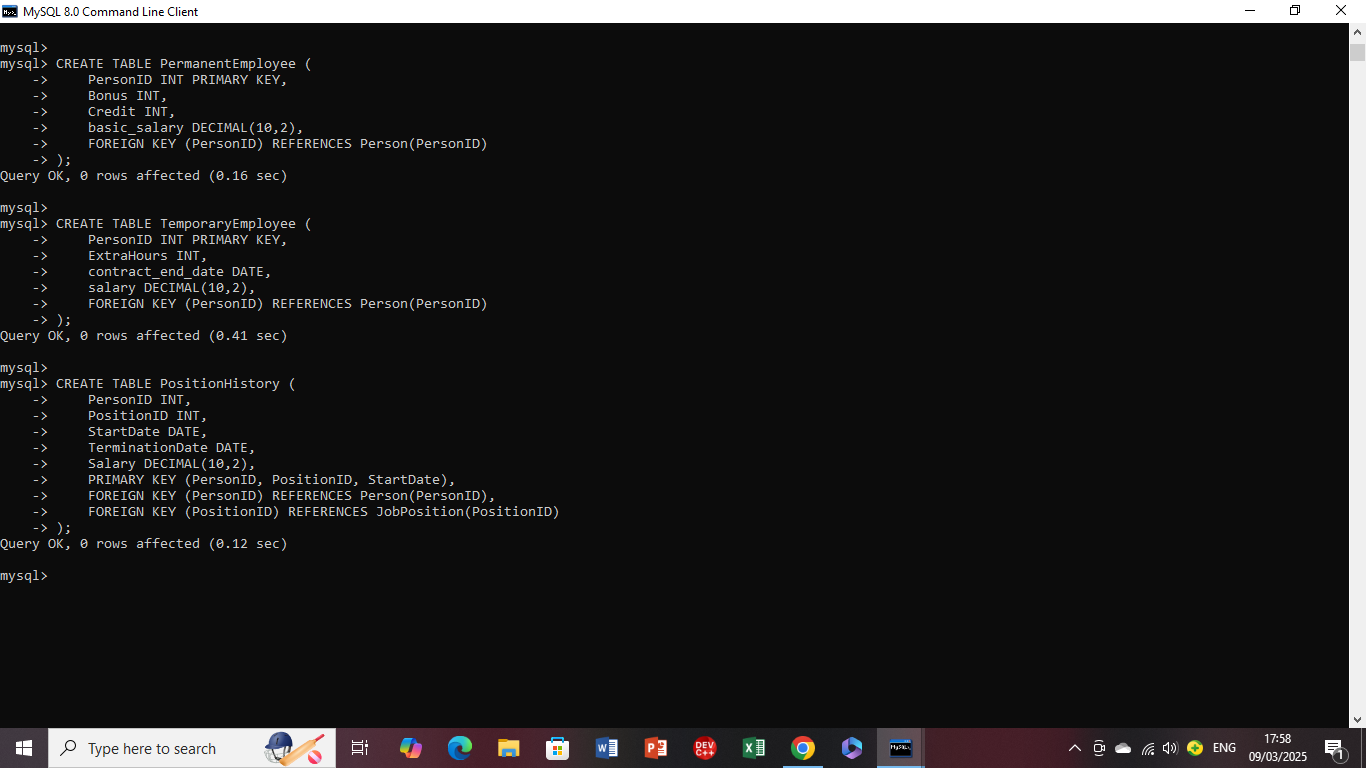
**Fig. 1**

## System Implementation/Prototyping

The database was implemented using MySQL. It involved:

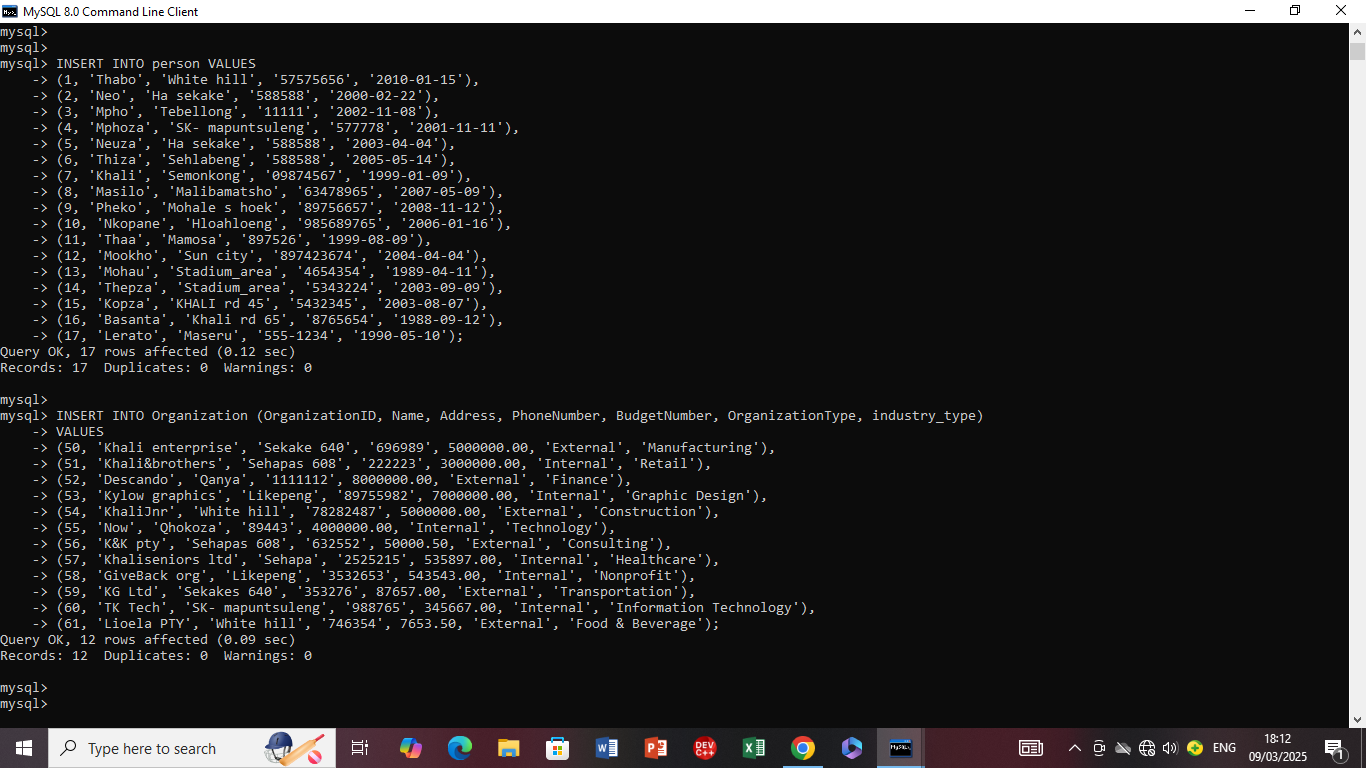
1. Creating the TT\_Holdings database using the command: CREATE DATABASE TT\_Holdings;
2. Selecting the database: USE TT\_Holdings;
3. Creating each of the seven tables (Person, Organization, Employment, Position, PositionHistory, PermanentEmployee, TemporaryEmployee) using the CREATE TABLE command

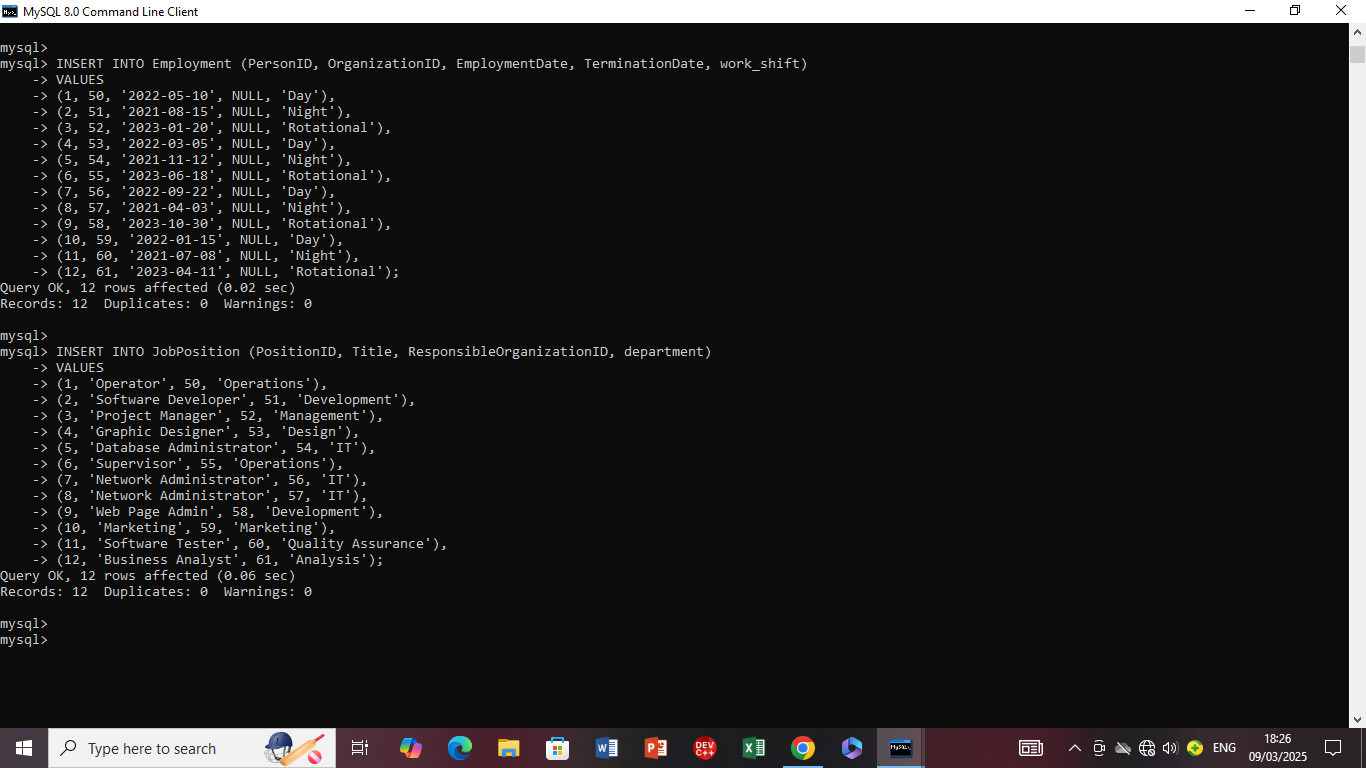
**Fig 2**



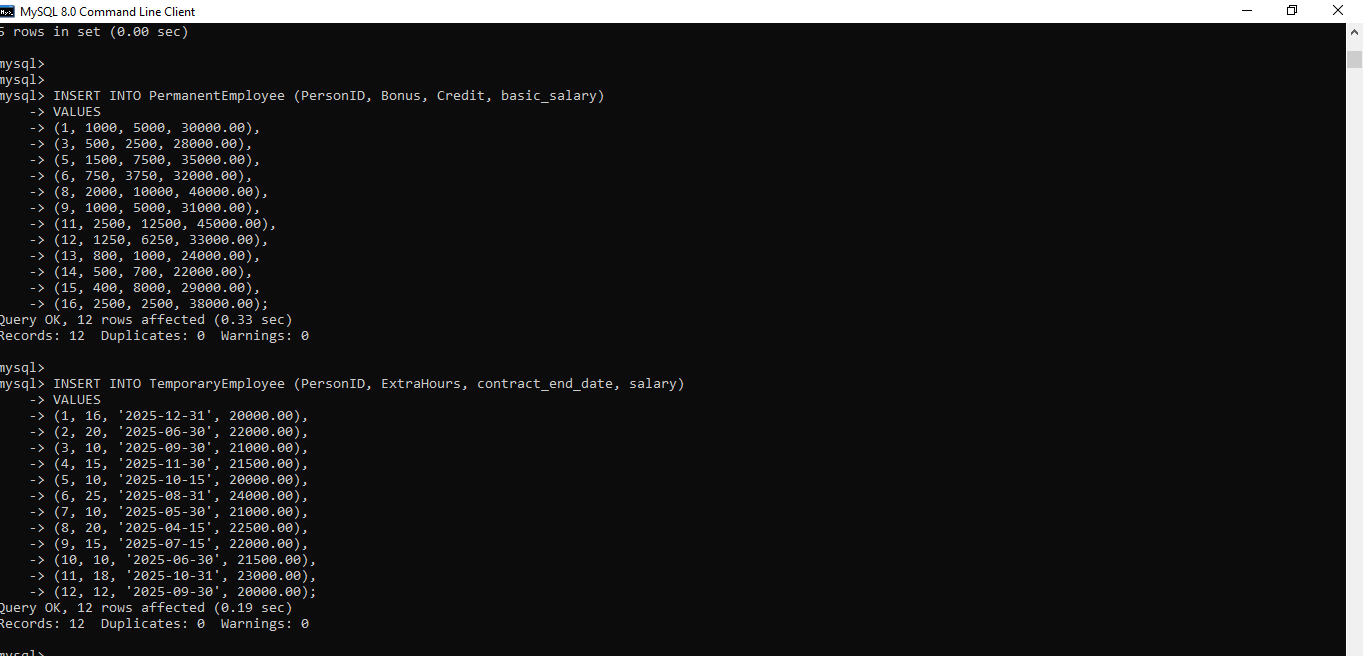
**Fig 3**

1. Inserting data into each table, ensuring that each table contained between 10 and 20 rows to represent realistic data scenarios.

 **Fig 4**

****

**Fig 5**

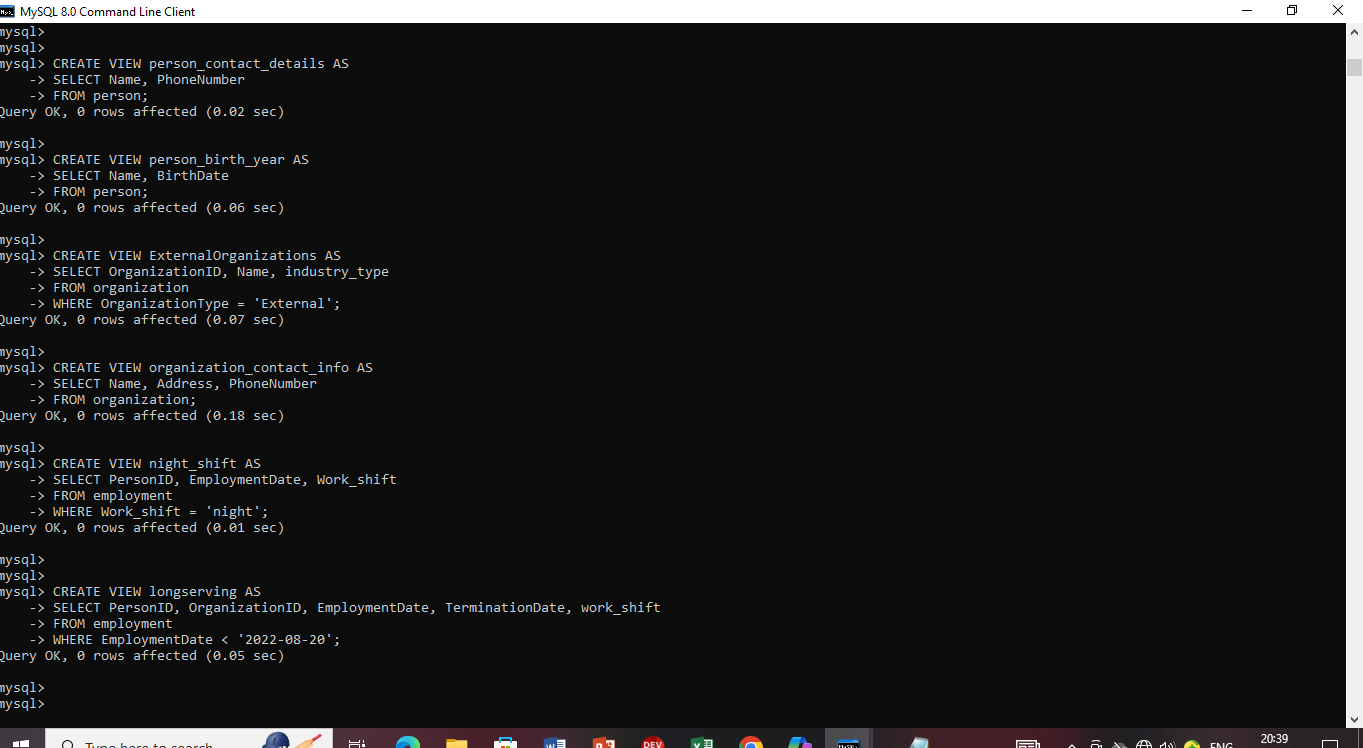
****

**Fig 6**

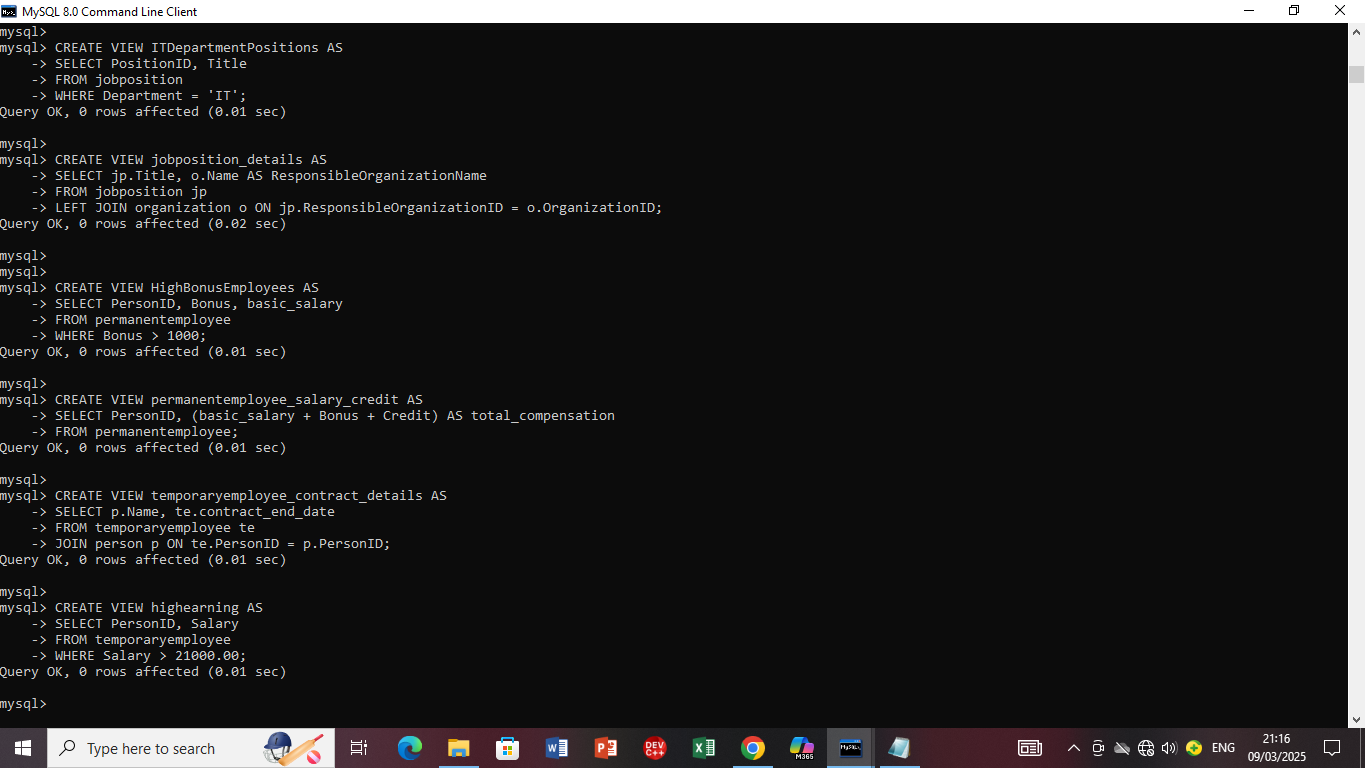


**Fig 7**

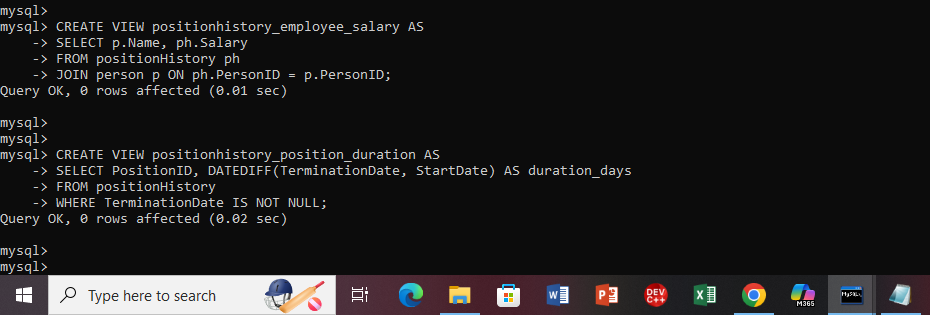
1. **Creating Views:** Two views were created for each table to simplify data access for specific reporting needs. CREATE VIEW command was used for this purpose.



**Fig 8**

****

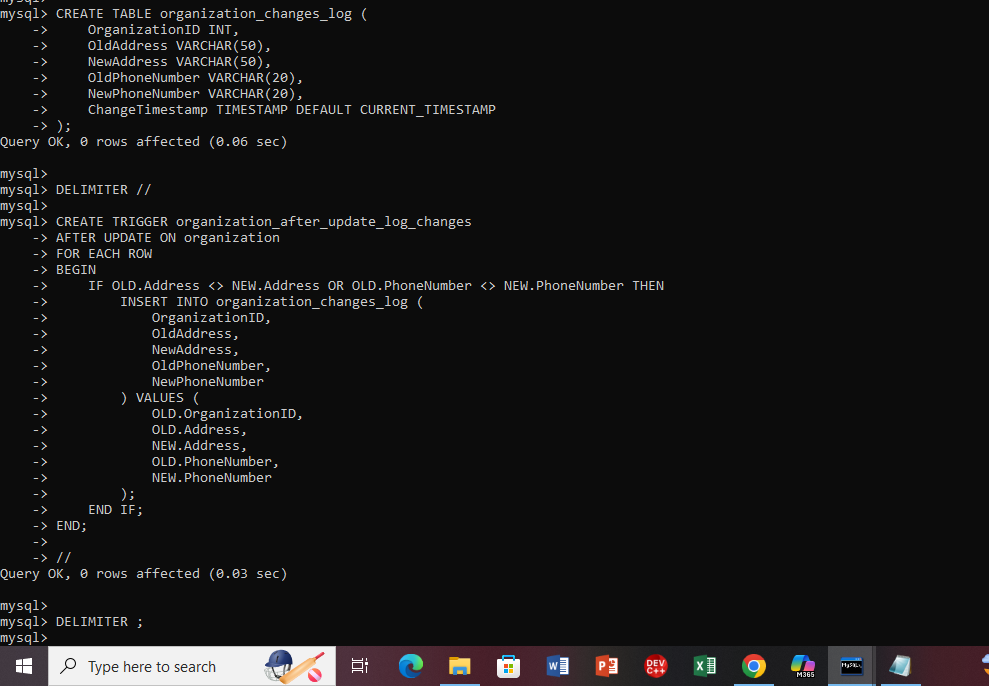
**Fig 9**



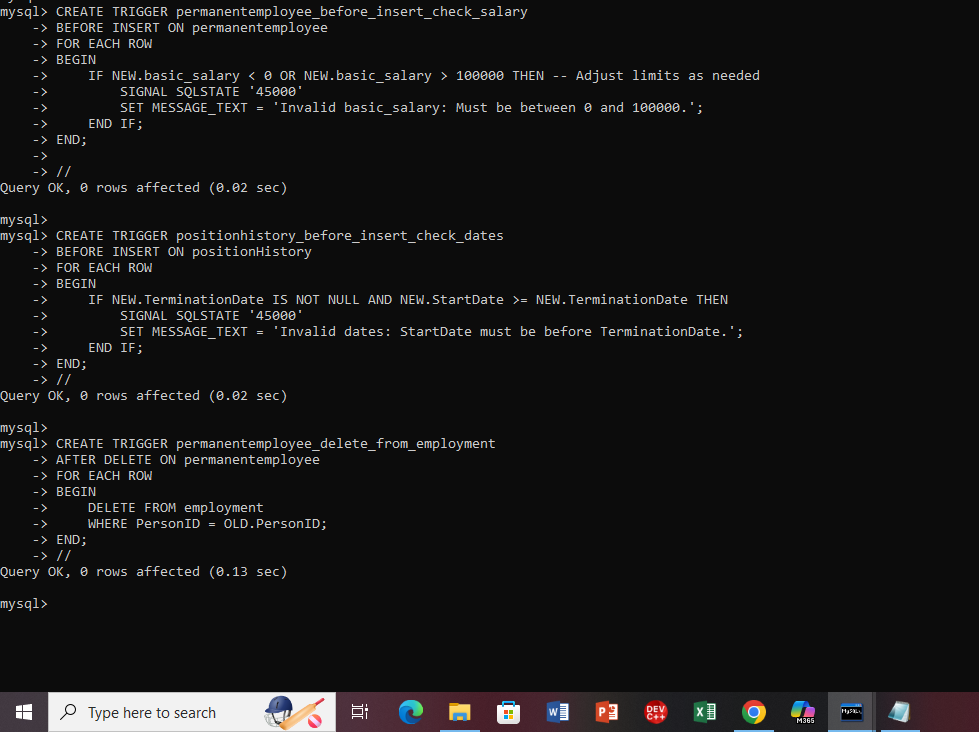
**Fig 10**

A view enables you to get structured data with a straightforward SELECT \* FROM view\_name; query, eliminating the need to repeatedly query numerous tables.This increases security by preventing unauthorized users from seeing specific columns.

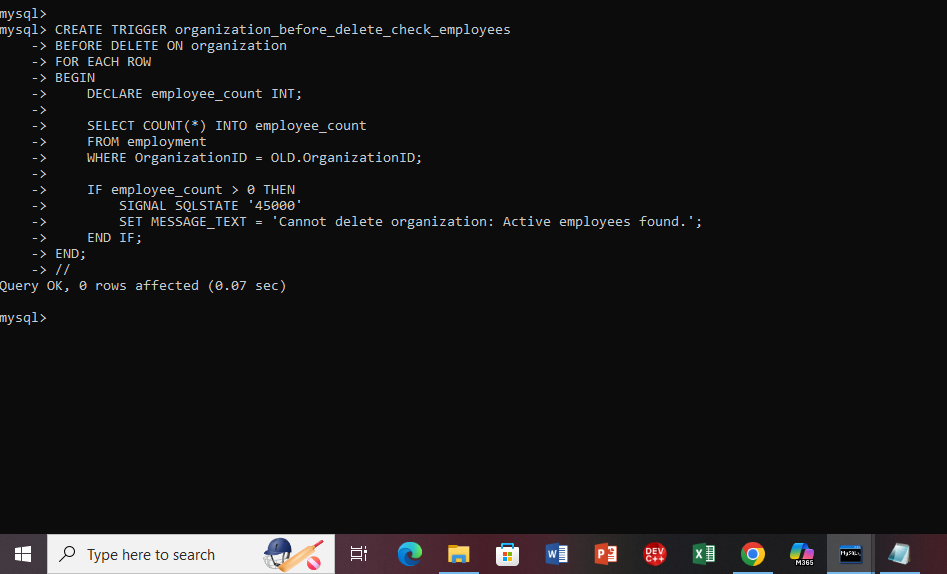
1. **Triggers:** Two delete, update, and insert triggers were implemented to enforce business rules and maintain data integrity.

****

**Fig 11**

****

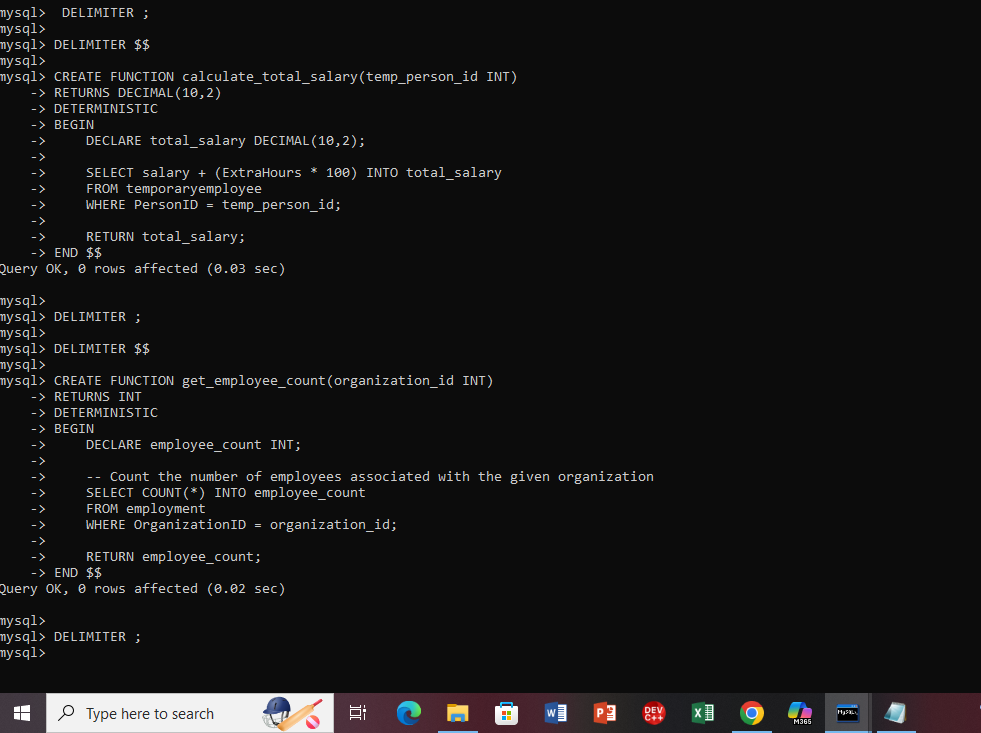
**Fig 12**

****

**Fig 13**

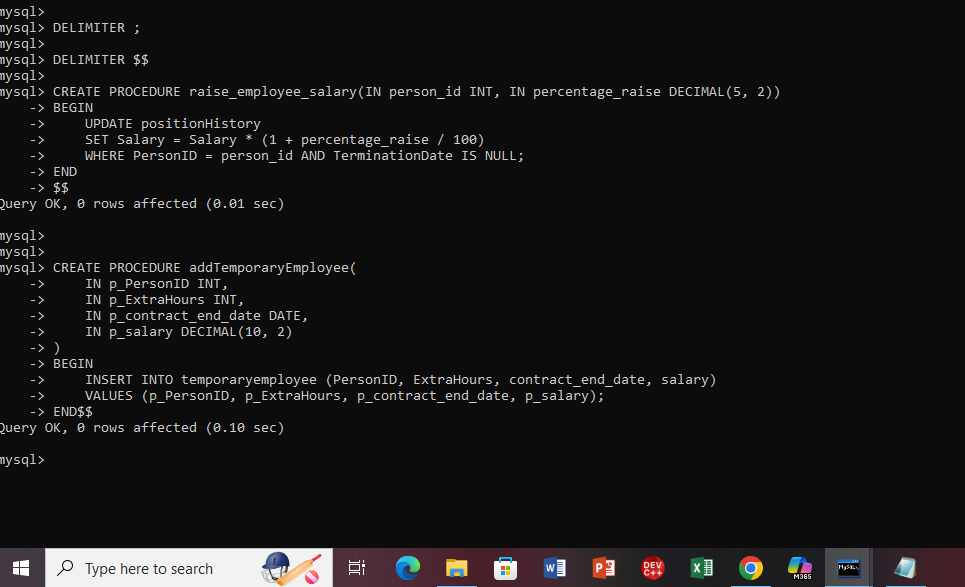
An event can occur before or after a trigger. Triggers make sure that no inaccurate information is entered and that changes in employment are automatically recorded.

1. **Functions:** Two functions were created to perform specific data calculations or transformations.

** Fig 14**

A function standardizes these computations rather than having to write them by hand each time.A function that returns the calculated salary is called, such as SELECT CalculateSalary(1001);.

1. **Stored Procedures:** Two stored procedures were implemented to encapsulate complex database operations.



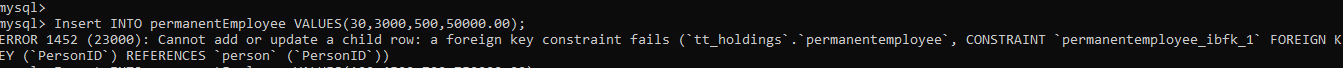
**Fig 15**

A procedure combines all of the actions into a single, reusable command rather than doing individual queries for each action. Again byy granting ccess to particular activities without disclosing table structures, procedures improve security.

## Testing

### Data Insertion Testing

On figure 3.3 the data was successfully inserted into table PermanentEmployee, this is because the data types and constraints were respected. To test the foreign key constraint between PermanentEmployee and Person, I tried to insert a record into PermanentEmployee with a PersonID that did not exist in the Person table. The attempt resulted with an error.



**Fig 16**

# Chapter 4 : System Initiation and Planning

## Assessing Project Feasibility

* **Technical feasibility**

The TT Holdings system is built using MySQL for the database, it is well-suited for handling structured data and user interactions.

* **Economic feasibility**

The system use MySQL which is an open-source technology, which reduces development costs because there is no need for additional licensing fees, making the project economically viable.

* **Operational feasibility**

The relational model provides a clear and understandable structure for managing the data. The design considered the need to manage different types of employees and organizational structures, aligning with the likely operational requirements of TT Holdings.

## Project Plan

* **Requirement Gathering and Analysis**

As discussed in methodology the first step involves a thorough analysis of TT Holding database design which included identifying the key entities with their associated attributes.

* **Conceptual Design**

Following the though examination of TT Holding database the ER diagram was drawn to visually represent the database

* **Logical Design**

The conceptual ER diagram was mapped into relational schema defining the tables, columns, data types, primary keys, and foreign keys.

* **Data Insertion**

The data ranging from 10 to 20 rows per table, was inserted into the database using INSERT INTO statements to populate the tables with realistic data.

* **Advanced SQL implementation**

Advanced SQL features were used to preserve data integrity and enhance database performance. This required constructing two functions and two stored procedures, as well as two views for every table and two triggers for insertions, updates, and removals.

* **User Creation and Privilege Granting**

Two standard users were established in the database, with each user being assigned two specific privileges that align with their potential roles.

* **Documentation**

The final phase involved documenting the entire project process, findings, and implementation details in this report.

# Chapter 5 : System Analysis

These requirements are categorized into functional and non functional requirements

## Determining System Requirements

**Functional Requirements**

These list the fundamental tasks that the system must perform.

1. **Person management**
   * Preserve personal data such as ID, name, address, phone number, birthdate.
   * A person may be classified as both a permanent and a temporary employee
2. **Organization management**
   * Keep track of the organization's ID, name, address, phone number, and budget number.
   * A company may have a number of employees.
3. **Tracking Employment**

* Every job record links an individual to a company.
* The date of employment, the date of termination, and any incentives received are all included in employment records.

1. **Position management**

* Maintain a record of the position's specifics (title, related organization).
* A single post may be held by several people at various times.

1. **Position History**

* Keeping track of a person's employment history, including start and end dates and pay.
* Over time, a person may move between roles in the same company or different ones.

1. Employee classification

* PermanentEmployee: Eligible for bonuses and credits.
* TemporaryEmployee: Hours worked beyond the regular schedule are documented.

**Non Functional requirements**

* **Security:** only authorized individuals should have access to sensitive employee and organization data
* **Scalability:** System ought to be able to handle an increasing number of employees and organizations.

## Structuring System Requirements

**Modules of the System**

* **Person module:** Manages personal details and employment type
* **Organization module:** Holds information about businesses that hire people.
* **Employment module:** Monitors the beginning and ending dates of employment
* **Position management module:** manages job titles, start and end dates and related compensation data.

**Structure of ER model**

* Person (ID, Name, Birthdate, Address, Phone)
* Organization (ID, Name, Address, Phone, BudgetNumber)
* Employment (EmploymentID, PersonID, OrganizationID, StartDate, EndDate, Bonus)
* Position (PositionID, Title, ResponsibleOrganizationID)
* PositionHistory (ID, PersonID, PositionID, StartDate, EndDate, Salary)
* PermanentEmployee (PersonID, Bonus, Credit, Basic\_Salary)
* TemporaryEmployee (PersonID, ExtraHours, Contract\_end\_date, Salary)

**Relationships**

The entities are connected in the following ways:

* A Person can have many Employments with different Organizations.
* An Organization can have many Employments with different Persons.
* An Employment can lead to a Person holding many Positions over time (tracked in Position History).
* A Position can be held by many Persons at different times (tracked in Position History).
* A Position is the responsibility of one Organization.
* A Person can be either a Permanent Employee or a Temporary Employee, with extra details stored in those specific tables. The PersonID links them.

# Chapter 6 : Conclusion

## Advantages of the System

A number of significant benefits are offered by the TT Holding Comprehensive Employment Management System:

* Data Consistency & Integrity: All job, position, and organization data are constently and methodically kept with referential integrity thanks to the relational database schema
* Effective Employee Monitoring: The system makes it possible to monitor past employment, including changes in job and pay information.
* Automation with SQL Features: Stored procedures automate complex queries, triggers enforce business rules, and views streamline data retrieval.
* User Privileges & Security: To ensure restricted access to sensitive data, distinct users are given various privileges.

## Future Enhancement of the System

Future improvements may include:

* Web Application: A front-end application to allow interaction with the system more intuitively.
* Creating automated reports on hiring trends, pay expenditures, and personnel trends is known as data analytics and reporting.

## Potential Benefit

TT Holdings can gain a lot from the deployment of this technology, including:

* Better Decision-Making: Management can decide on recruiting, promotions, and workforce distribution with the use of organized and up-to-date data.
* Improved Compliance: The organized database guarantees that company policies and employment laws are followed.

## Conclusion

The necessity for organized personnel and organizational management is effectively met by the TT Holding Comprehensive Employment Management System. This system makes sure that employment records are kept effectively and safely by utilizing relational database design, sophisticated SQL functions, and data integrity procedures. The system will become an invaluable asset for TT Holdings as a result of future improvements that will further strengthen its automation capabilities, security, and usability.

# References

Chen, W., 2023. *Database Design and Implementation.* [Online]   
Available at: https://orc.library.atu.edu/atu\_oer/2  
[Accessed 25 03 2025].

Gupta, P., Agarwal, R. & Verma, A, 2016. SQL Optimization Techniques for Employee Databases. *Journal of Computing Research,* 12(2), pp. 112-128.

Huang, L., Park, J. & Kim, D, 2018. Enhancing Employment Database Security Using SQL Views. *Journal of Information Systems Security,* 10(4), pp. 233-247.

Mullins, C., 2012. *Database Adminstration,* United Kingdom: Pearson Education.

# Appendices

## Appendix A : SQL code

### Table creation

These are the statements for creating tables that were mentioned in List of tables:

create table Person (

PersonID INT PRIMARY KEY,

Name varchar(50),

Address varchar(50),

PhoneNumber varchar(20),

BirthDate DATE

);

create table Organization (

OrganizationID INT PRIMARY KEY,

Name varchar(50),

Address varchar(50),

PhoneNumber varchar(20),

BudgetNumber DECIMAL(10, 2)

);

create table Employment (

PersonID INT,

OrganizationID INT,

EmploymentDate DATE,

TerminationDate DATE,

Work\_shift enum('Day','Night','Rotational'),

PRIMARY KEY (PersonID, OrganizationID),

FOREIGN KEY (PersonID) REFERENCES Person(PersonID),

FOREIGN KEY (OrganizationID) REFERENCES Organization(OrganizationID)

);

CREATE TABLE JobPosition (

PositionID INT PRIMARY KEY,

Title VARCHAR(50),

ResponsibleOrganizationID INT,

Department VARCHAR(50),

FOREIGN KEY (ResponsibleOrganizationID) REFERENCES Organization(OrganizationID)

);

create table PositionHistory (

PersonID INT,

PositionID INT,

StartDate DATE,

TerminationDate DATE,

Salary DECIMAL(10, 2),

PRIMARY KEY (PersonID, PositionID, StartDate),

FOREIGN KEY (PersonID) REFERENCES Person(PersonID),

FOREIGN KEY (PositionID) REFERENCES JobPosition(PositionID)

);

CREATE TABLE PermanentEmployee (

PersonID INT PRIMARY KEY,

Bonus INT,

Credit INT,

Basic\_salary INT,

FOREIGN KEY (PersonID) REFERENCES Person(PersonID)

);

CREATE TABLE TemporaryEmployee (

PersonID INT PRIMARY KEY,

ExtraHours INT,

Salary INT,

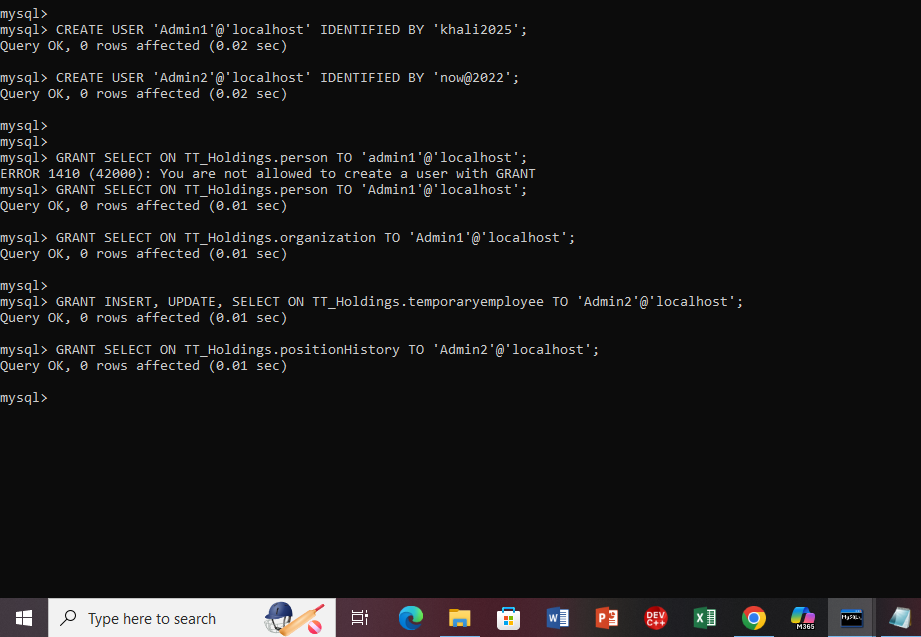
FOREIGN KEY (PersonID) REFERENCES Person(PersonID)

);

### Triggers

The screenshots for creation of triggers were inserted earlier on chapter 3 under system implementation.

## Appendix B: User privileges



Admin1 user has access to organizational data, employment information, and personnel records. They are authorized to add new employment records but cannot edit other employees’ data beyond their assigned rights.

Admin2 user is an administrator and is in charge of managing the entire database. They are able to add, edit, and delete records from every table. They can execute functions and stored procedures for automated tasks thanks to the EXECUTE privilege