Contents

[1.Title of the System 3](#_Toc194003696)

[2.list of tables 4](#_Toc194003697)

[5.List of Abbreviations 4](#_Toc194003698)

[5.Abstract 5](#_Toc194003699)

[6.Chapter 1: Introduction 5](#_Toc194003700)

[Problem Statement 6](#_Toc194003701)

[Problem Solving 6](#_Toc194003702)

[Objectives 7](#_Toc194003703)

[Scope & Constraints 7](#_Toc194003704)

[7.Chapter 2: Literature Review 8](#_Toc194003705)

[2.1 Introduction 8](#_Toc194003706)

[2.2 Reviewing the Literature 8](#_Toc194003707)

[2.3 Findings & Discussion 9](#_Toc194003708)

[8. Chapter 3: Methodology 10](#_Toc194003709)

[3.1 Requirement Analysis 10](#_Toc194003710)

[3.2 System Design 11](#_Toc194003711)

[3.3 System Implementation / Prototyping 12](#_Toc194003712)

[3.4 Testing 14](#_Toc194003713)

[3.5 Documentation 14](#_Toc194003714)

[9. Chapter 4: System Initiation and Planning 15](#_Toc194003715)

[4.1 Assessing Project Feasibility 15](#_Toc194003716)

[4.2 Project Plan 16](#_Toc194003717)

[10.Chapter 5: System Analysis 17](#_Toc194003718)

[5.1 Determining System Requirements 17](#_Toc194003719)

[5.2 Structuring System Requirements 19](#_Toc194003720)

[11.Chapter 6: Database Design 20](#_Toc194003721)

[6.1 ER/EER Diagram 20](#_Toc194003722)

[6.2 Mapping 21](#_Toc194003723)

[6.3 Database Creation 22](#_Toc194003724)

[6.4 Data Insertion 22](#_Toc194003725)

[6.5 Advanced SQL (Views, Triggers, Functions, Procedures) 23](#_Toc194003726)

[6.6 User Creation and Privileges 24](#_Toc194003727)

[12.Chapter 7: Conclusion 25](#_Toc194003728)

[13. References 26](#_Toc194003729)

[16.Appendices of 28](#_Toc194003730)

# 

# 1.Title of the System

**TT Holding Database design**

# 2.list of tables

1.Person

2. Organization

3. Employment

4. Position

5. EmploymentPosition

6. PermanentEmployee

7. PartTimeEmployee

8. OrganizationType

9. BonusDetails

10. SalaryDetails

# 5.List of Abbreviations

**- DB**: Database

**- DBMS**: Database Management System

**- ER:** Entity-Relationship

- **EER**: Enhanced Entity-Relationship

**- HR**: Human Resources

**- SQL**: Structured Query Language

- **UAT**: User Acceptance Testing

- **RDBMS**: Relational Database Management System

- **FK**: Foreign Key

- **CRUD:** Create, Read, Update, Delete

- **API**: Application Programming Interface

# 5.Abstract

This report presents the design of a database management system for TT Holding, aimed at addressing challenges in managing employee information across multiple organizations.AS mentioned by Navathe (2016), in today’s dynamic work environment, organizations often find it difficult to track employee histories, positions, and organizational affiliations. This database seeks to offer a structured solution for storing, managing, and retrieving critical data related to employees and their employers. The proposed system captures important details such as employee IDs, names, addresses, phone numbers, birth dates, and organizational budgets. According to Rob (2016) it allows for differentiation between permanent and part-time employees while keeping track of bonuses, hours worked, and responsibilities for each role. The system has been designed using Entity-Relationship (ER) and Enhanced Entity-Relationship (EER) diagrams to ensure scalability and maintain data integrity as mentioned by Date (2004).

Advanced SQL techniques, including the creation of views, triggers, functions, and stored procedures, have been employed to optimize database performance and maintainability. The methodology includes requirement analysis, architectural design, and implementation, followed by testing to validate the system’s overall functionality. The resultant database management system is expected to enhance TT Holding’s ability to manage employee data effectively, thereby improving operational efficiency and decision-making as discovered by Kendall (2013). This report outlines the design process, implementation strategies, and the anticipated benefits from adopting this database system.

# 6.Chapter 1: Introduction

**Background**

In recent years, organizations have increasingly recognized the pivotal role that effective human resource management plays in achieving overall business success. Human resources (HR) is not just about managing employee records but also involves strategic planning, talent acquisition, performance management, and compliance with various labor laws as mentioned by Coronel (2016). As businesses grow and diversify, especially in a multi-entity corporate environment like TT Holding, the complexity of managing employee information escalates. Traditional methods of maintaining employee records—often carried out manually or in isolated systems—result in inefficiencies, data discrepancies, and increased risks of non-compliance. As mentioned by Navathe (2015). Consequently, HR departments face significant challenges in accessing timely and accurate information, thereby hindering their ability to make informed decisions. This project seeks to address these pressing issues by proposing a centralized database management system to streamline employee record management across TT Holding’s entities.

## Problem Statement

TT Holding, which operates multiple distinct entities, faces significant challenges in managing employee information due to a lack of a centralized system. According to Date (2004), Each entity maintains its own records, leading to fragmentation, inefficiencies, and difficulties in data retrieval. As a result, HR professionals grapple with prolonged data processing times, inaccuracies in employee records, and challenges in ensuring compliance with labor regulations. These issues not only hinder decision-making but also diminish overall operational efficiency, highlighting the urgent need for a structured and cohesive database management system.

## Problem Solving

To address these challenges, the proposed solution is to develop a centralized database management system specifically tailored for TT Holding's employee records. This system will integrate information from various entities into a single platform that can streamline processes such as hiring, onboarding, and performance management as mentioned by Elmasri (2015). By utilizing SQL for efficient data storage and retrieval, HR personnel will be able to access accurate, up-to-date employee information in a timely manner. Rob (2016) stated that this approach is expected to reduce data errors significantly and enhance compliance with labor laws, ultimately leading to improved employee satisfaction and organizational efficiency.

## Objectives

**The primary objectives of this project are as follows:**

**Database Design**: To create a robust database that captures essential employee data, including personal details, employment history, and job roles.

**Enhanced Functionality**: To implement advanced SQL techniques to optimize the database's functionality for efficient data access and management.

**Data Integrity and Security**: To ensure that the database maintains high standards of data integrity and security through effective architecture and management practices.

**User Accessibility**: To provide an intuitive interface for HR personnel, facilitating easy access to employee information and improving workflow efficiency.

By fulfilling these objectives, the project aims to create a comprehensive solution that addresses existing challenges in employee management at TT Holding.

## Scope & Constraints

**The scope of this project encompasses the following key areas:**

-**Data Definition**: Identifying the critical data points necessary for managing employee information and structuring them within the database framework.

**-Database Architecture:** Developing an Entity-Relationship (ER) diagram to illustrate the relationships between the various data entities.

-**Implementation:** Utilizing Structured Query Language (SQL) to design and create database tables, establish relationships, and enforce constraints to ensure data integrity.

**-Testing and Evaluation**: Conducting thorough testing of the system to guarantee its functionality and efficiency upon implementation.

**However, the project will face certain constraints, including:**

-**Limitations on User Interface Design:** The project will focus solely on back-end database functionalities and will not delve into front-end user interface aspects.

**-Resource Limitation**: Availability of technical resources and expertise may impact the implementation timeline and complexity of the database system.

# 7.Chapter 2: Literature Review

## 2.1 Introduction

In today's fast-paced business world, managing human resources effectively is more important than ever. With tech advancements, organizations are turning to centralized database management systems (DBMS) to keep track of their employee information. This chapter reviews existing literature on the benefits and challenges of implementing such systems in HR management. It will also indicate where you can incorporate screenshots of your database to provide a visual representation of the concepts discussed.

## 2.2 Reviewing the Literature

**The Role of Centralized Databases in HR Management**

Centralized databases play a crucial role in how organizations manage human resources today. According to scholars like Navathe (2015), having a single, centralized system for HR data makes it easier to ensure that all information is accurate and up-to-date. This is particularly vital in HR, where decisions often rely on precise employee data.

**Challenges in Implementation**

Despite the many benefits, implementing a centralized HR database is not without its challenges. One major issue is resistance from employees. Many people feel comfortable with existing processes and may be resistant to changing their routines. Date (2004) points out that this kind of resistance can hinder the transition to a new system.

Another concern is data security. Centralizing sensitive employee information can increase the risk of data breaches if not managed correctly. Navathe (2015) emphasize the importance of establishing strong policies to protect this data. Screenshots demonstrating security features or data governance options can illustrate how your database addresses these concerns.

**Frameworks for Effective Implementation**

To navigate the challenges of implementing a centralized database, frameworks and strategies are essential. One helpful approach is Kotter’s 8-Step Change Model, which guides organizations through the change process by focusing on factors like urgency and employee engagement as mentioned by Rob (2016).

## 2.3 Findings & Discussion

From the literature, it’s clear that a centralized database can significantly improve HR management. It helps maintain accurate employee records, which is crucial for making informed decisions. The ability to access consolidated information quickly can enhance how HR departments function, leading to smoother operations and better compliance with regulations. However, the findings also highlight that transitioning to a centralized system requires effective change management strategies. Employees can be hesitant to adopt new technologies, so it's essential to communicate the advantages clearly and offer adequate training.Data security remains a top priority. Organizations need to ensure that the sensitive information within these databases is well-guarded. Effective data governance policies are vital for maintaining trust and transparency among employees.

In conclusion, while implementing a centralized database brings several challenges, the benefits can far outweigh them if approached thoughtfully. By using frameworks for change management and focusing on data security, organizations like TT Holding can harness the full potential of a centralized DBMS.

# 8. Chapter 3: Methodology

## 3.1 Requirement Analysis

The first phase of the project involved a comprehensive analysis of the system requirements based on the defined business scenario. This included identifying all necessary data entities and their relationships. The key entities established include:

**- Person:** Represents individuals who may be employed by one or more organizations.

**- Organization:** Represents internal and external entities that employ persons.

**- Employment:** Captures the employment relationship between persons and organizations, including details like employment dates and bonuses.

**- Position:** Represents job titles that can be held within organizations.

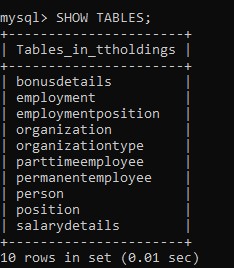
**- Bonus Details:** Captures data about various bonuses awarded to employees.

According to Sommerville (2011). The gathering of requirements was facilitated through discussions with stakeholders to ensure full alignment with business needs.

## 3.2 System Design

**Architectural Design**

The architecture for the TT Holding database was crafted as a relational database model to ensure data consistency and integrity. Key components of the architecture include the following tables in a picture.



An Entity-Relationship (ER) diagram (to be included in the Appendices) visualizes the structure, highlighting the relationships such as:

- A person can be employed by multiple organizations.

- Each organization can employ several persons.

- A person can hold multiple positions over time.

**UML Diagrams**

Unified Modeling Language (UML) diagrams were utilized to represent system interactions and flows extensively. Class diagrams, sequence diagrams, and activity diagrams were created to illustrate:

- The relationships between entities.

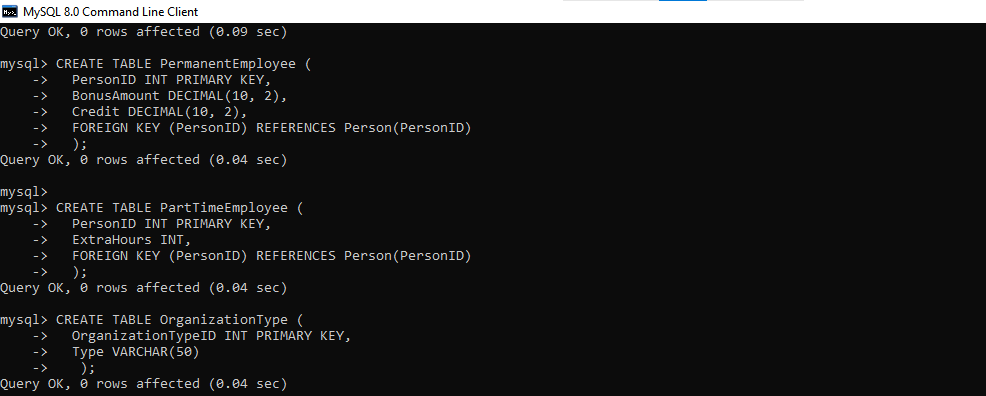
- Processes such as hiring and employee management.

- Reporting needs for different stakeholder roles.

## 3.3 System Implementation / Prototyping

**Database Creation**

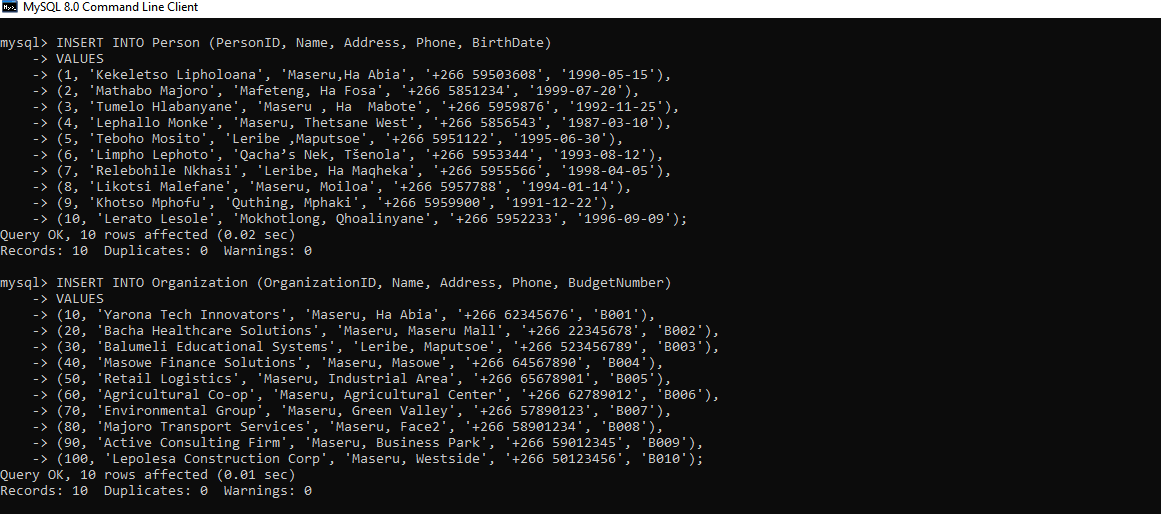
The database was created using MySQL. Below are the SQL commands used for creating essential tables:



Similar commands were used to create other tables. After creating the tables, a total of 15 rows were included in the Person table, alongside similar data populations in others such as Organization and other tables.

**Data Insertion**

Data insertion scripts were executed to populate each table, ensuring the dataset was sufficient for testing and operational purposes. The following diagram demonstrates the insertion of data into the Person table:



**Advanced SQL Features**

The implementation phase also included the development of SQL views, triggers, functions, and procedures:

**- Views:** Created views to simplify data access, aggregating key information about persons, organizations, and employment statuses.

**- Triggers:** Implemented to maintain audit logs for actions taken on important entities, ensuring data integrity and history tracking.

-**Functions and Procedures:** Custom functions (e.g., calculating age) and procedures (e.g., fetching employment details) were developed to enhance the database's usability.

## 3.4 Testing

To ensure reliability, the database underwent several testing phases:

- **Unit Testing:** Each table and relationship was verified for accurate data representation and integrity.

**- Integration Testing**: The functionality of SQL scripts for views, triggers, and procedures was assessed.

**- User Acceptance Testing (UAT):** Real-world scenarios were simulated to confirm that all functionalities met user requirements and that stakeholders could perform their tasks efficiently.

## 3.5 Documentation

Extensive documentation was compiled throughout the project to record the processes and methodologies applied. This material covered:

- ER/EER diagrams

- Mapping of relationships

- SQL script documentation for table creation, data insertion, advanced queries, etc.

- User manuals to facilitate training for end-users on how to use the new database effectively.

# 9. Chapter 4: System Initiation and Planning

## 4.1 Assessing Project Feasibility

The feasibility assessment for the TT Holding database project was conducted to evaluate its viability across technical, economic, and operational dimensions.

**Technical Feasibility**

This aspect assessed the compatibility of existing technology with the proposed database system:

-**Infrastructure Compatibility:** TT Holding's current IT infrastructure was evaluated, confirming its capacity to support an RDBMS like MySQL.

- **Technical Skills:** The project team possessed the requisite skills in database design and SQL programming. Training was provided to fill any knowledge gaps on MySQL as mentioned by Kendall (2013).

**Economic Feasibility**

This evaluation analyzed the project’s cost-effectiveness and expected benefits:

-**Cost Analysis:** Initial costs included software licenses, development, and training. A budget was created for effective financial tracking.

-**Benefits Analysis:** Forecasted benefits included improved data access and enhanced employee management processes. A cost-benefit analysis indicated that benefits would surpass costs within a specified timeframe as stated by Bourne (2018).

**Operational Feasibility**

This dimension assessed user acceptance and stakeholder involvement:

**- Stakeholder Involvement:** HR, finance, and operations staff provided input during the requirements phase, ensuring the system met user needs.

**- Change Management**: According to Kotter (2007). A comprehensive change management plan was developed to mitigate resistance and included training and support strategies.

The assessments concluded that the project is technically, economically, and operationally feasible.

# 4.2 Project Plan

The project plan outlines the timeline, tasks, and resource allocation required for the TT Holding database system's implementation. A Gantt chart detailing project milestones is provided in the appendices.

**Project Timeline**

The project is structured into key phases, as follows:

1. **Project Initialization (Weeks 1-2):** Define scope, assemble the team, and hold initial meetings.

2. **Requirements Gathering (Weeks 3-4):** Conduct stakeholder interviews, document requirements, and obtain approvals.

3**. System Design (Weeks 5-6**): Create database schema and user interface prototypes.

4. **Implementation (Weeks 7-10):** Develop the database, set up triggers and views, and add test data.

5. **Testing (Weeks 11-12):** Conduct unit and user acceptance testing.

6. **Deployment (Week 13):** Roll out the system and provide training.

7. **Post-Implementation Support (Weeks 14-15):** Monitor performance and adjust based on feedback.

**Resource Allocation**

Effective resource allocation is paramount for project success:

**-Project Manager:** Oversees execution and communication.

- **Database Developers:** Design the database structure.

- **Business Analysts:** Gather and align requirements.

**- QA Specialists**: Validate functionalities through testing.

**- Training Facilitators:** Conduct training sessions.

A server environment will be provisioned to host MySQL and any necessary business intelligence tools.

**Risk Management**

Identifying potential risks ensures project stability:

- Scope Creep: Regular reviews are conducted to document changes.

- Resource Availability: Team member availability will be monitored continuously.

- User Resistance to Change: Continuous stakeholder engagement and support will be prioritized.

The project feasibility analysis indicates that the TT Holding database project is viable and beneficial for enhancing organizational efficiencies. The structured project plan provides a roadmap for implementation, ensuring all critical steps are taken for a successful database solution.

# 10.Chapter 5: System Analysis

## 5.1 Determining System Requirements

Determining system requirements is essential for ensuring that the TT Holding Database meets user expectations and business needs. According to Kendall (2013) this process incorporates both functional and non-functional requirements gathered through stakeholder interviews, surveys, and workshops.

**Functional Requirements**

Functional requirements describe the specific behaviors and functionalities that the system must support. Key functional requirements for the TT Holding Database include:

**1. User Management:**

- Creation, modification, and deletion of user accounts for normal users and administrators.

- Assignment of roles and permissions based on job functions.

**2. Data Management:**

- Facilitation of creation, retrieval, updating, and deletion for data entities such as Person, Organization, Employment, Position, PermanentEmployee, and PartTimeEmployee.

- Support for batch data insertion and data export capabilities.

**3. Employment Records:**

- Recording management of employment details, including dates and bonuses linked to

each employment instance.

- Tracking changes in employment positions and salary information.

**4. Reports and Queries:**

- Generation of employee, organization, and employment trend reports.

- Support for advanced queries to analyze data relationships.

**5. Bonus Implementation:**

- Management of bonus details, including performance, retention, and signing bonuses awarded to employees.

**Non-Functional Requirements**

Non-functional requirements specify how the system performs certain functions and its quality attributes. Important non-functional requirements include:

1. **Usability**:

- A user-friendly interface with intuitive navigation for users of varying technical skills.

- Provision of online help features and user manuals as mentioned by Bourne (2018).

2. **Performance**:

- Capability to handle up to 100 concurrent users without degradation.

- Page loads within three seconds and query execution under two seconds for frequent requests.

**3. Security**:

- Strict access controls to safeguard sensitive data from unauthorized access.

- Secure storage of user data, including encryption of sensitive fields as mentioned by Kotter (2007).

**4. Scalability:**

- Architecture designed for easy scaling to accommodate data growth and increased user load.

- Support for future enhancements with minimal disruption.

**5. Backup and Recovery:**

- Automated backup processes to ensure restore capability during failures.

- Documented recovery procedures for system administrators.

## 5.2 Structuring System Requirements

To facilitate implementation, the identified requirements are organized into a structured document categorized as follows:

**Requirement Categories**

1**. User Requirements:** Address user expectations, including usability, accessibility, and tailored functionality for various roles.

**2. System Requirements**: Include hardware and software specifications necessary for operation, such as server specifications and required software versions.

**3. Data Requirements**: Define data entities, their relationships, data types, and constraints like primary and foreign keys.

**Use Case Diagrams**

Use case diagrams have been developed to visually represent user interactions with the system. Each functional requirement is mapped to specific use cases, illustrating user engagement with system components (diagrams provided in the appendices). Key use cases include:

**1. User Login and Authentication:** Role-based login and access.

**2. Managing Person Records:** CRUD (Create, Read, Update, Delete) operations on person records.

**3. Employment History Tracking:** Viewing and updating linked employment records.

**4. Generating Reports:** Report generation on employment data and bonuses.

# 11.Chapter 6: Database Design

## 6.1 ER/EER Diagram

Entity-Relationship (ER) and Enhanced Entity-Relationship (EER) diagrams are foundational tools in database design, serving as visual representations of entities in a system and their interrelations. In this project, the ER/EER diagram was constructed to illustrate the relationships between various entities such as Person, Organization, and Employment.

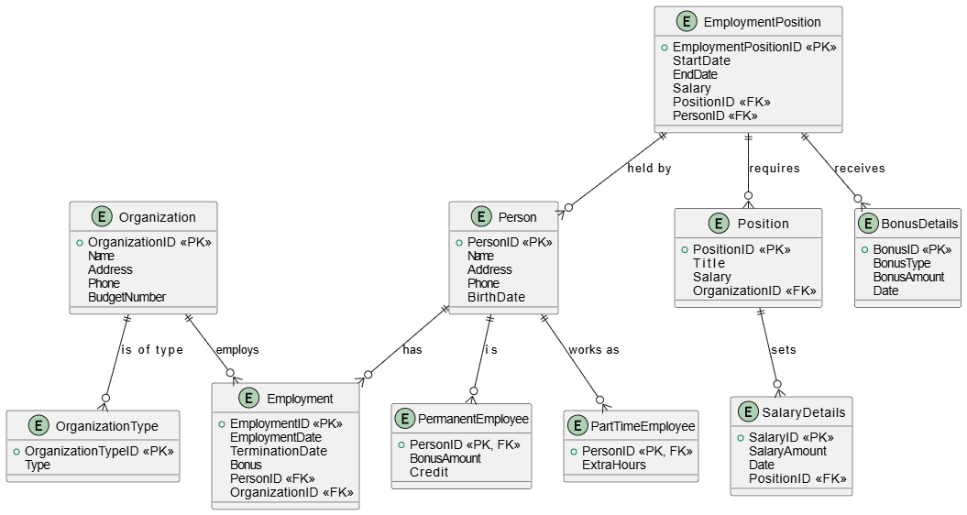
**The diagram outlines the following characteristics**:

- Entities and Attributes: Each entity is defined with specific attributes. For example, the Person entity includes attributes like PersonID, Name, Address, Phone, and BirthDate, while the Organization entity contains OrganizationID, Name, Address, Phone, and BudgetNumber.

- Relationships: The relationships between entities are depicted clearly, indicating how, for instance, a Person can have multiple Employment records and how each Employment record is associated with a particular Organization.

- Cardinality: The diagram specifies cardinality constraints, such as one-to-many relationships between Organization and Employment, and many-to-one relationships between Employment and Person as stated by Navathe(2016).

Figure 6.1 below demonstrates the ER/EER diagram constructed for the TT Holdings database.



## 

## 6.2 Mapping

Mapping the ER/EER diagram into a physical database schema is crucial for implementing the design in a relational database management system (RDBMS). According to Rob (2018) this process involves converting entities into tables, attributes into columns, and relationships into foreign keys.

**Mapping Process Details**

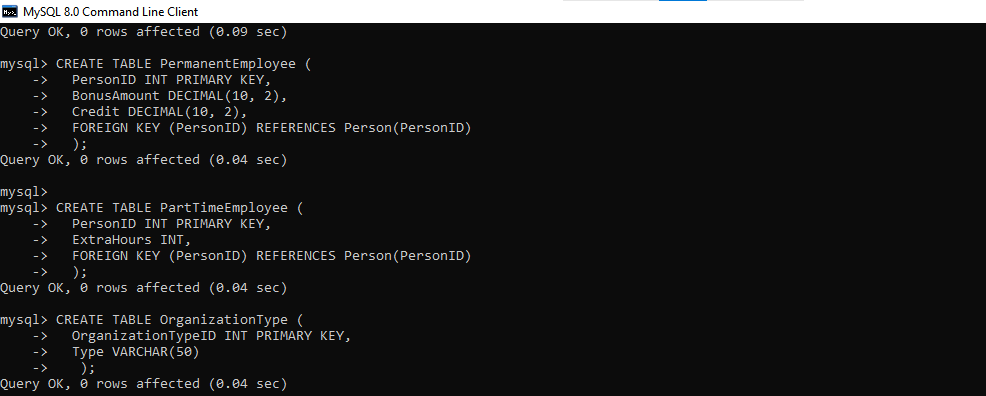
1. Entities Transformed into Tables: Each entity from the ER diagram, such as Person, Organization, and Employment, corresponds directly to a table in the database.

2. Attributes as Columns: The attributes associated with each entity are represented as columns within their respective tables. For example, the Person table has columns for PersonID, Name, Address, Phone, and BirthDate.

3. Relationships Using Foreign Keys: Relationships indicated in the ER diagram are implemented in the database schema through foreign keys. For instance, the Employment table includes a foreign key that references the PersonID from the Person table, establishing the relationship between employment records and employees.

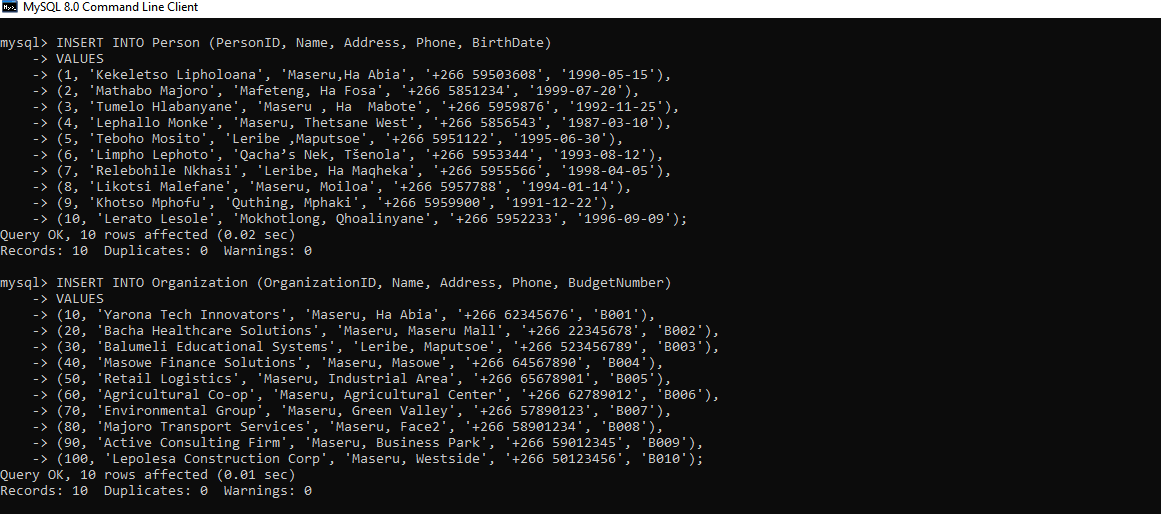
The mapping process generates a structured representation of the database that facilitates efficient data storage and retrieval.

## 6.3 Database Creation

Creating a database entails establishing the framework to house all related data structures. In this chapter, a database named TT\_Holdings was created, followed by the definition of tables to store specific types of data. Below is a summary of SQL commands used to create tables within the TT\_Holdings database:

## 6.4 Data Insertion

The next step involves populating the database tables with initial data. This is performed through structured SQL INSERT statements that add records into each table. By inserting sample data into our tables, we effectively simulate a working dataset that can be manipulated and queried as needed.

Example: INSERT command for the `Person` table  `:

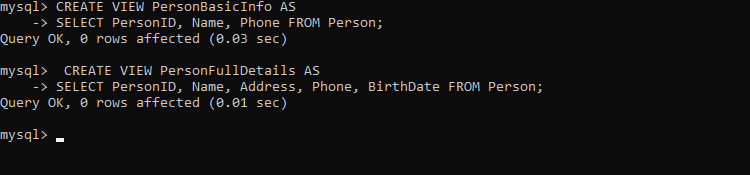
## 6.5 Advanced SQL (Views, Triggers, Functions, Procedures)

To enhance database functionality, advanced SQL techniques were implemented, including views, triggers, functions, and stored procedures.

**Views**

Views abstract complexity from end-users and present a simplified schema. For instance, views like PersonBasicInfo provide easy access to essential details without exposing the entire underlying table (Date, 2004).

The following figure shows how views are created.



**Triggers**

Triggers automatically respond to events in the database, aiding in maintaining integrity and logging actions. Key triggers were designed to log insertions and updates into the AuditLog table, ensuring a clear historical record of changes. How triggers designed is shown in the appendices.

**Functions**

Functions provide reusable code that can be invoked from SQL statements to perform tasks such as calculating employee ages. This encapsulation enables more manageable and modular code. Functions implementations are provided on the appendices.

**Procedures**

Stored procedures encapsulate database operations into callable routines, simplifying complex operations. For example, a procedure named Get\_Employment\_Details retrieves employment information based on an employee ID. Procedure implementations are provided on the appendices.

## 6.6 User Creation and Privileges

For effective database management and security, user accounts were created with specific privileges, ensuring only authorized individuals can perform certain actions. This enhances security while maintaining data integrity throughout all interactions with the database as mentioned by Navathe (2016). User creation and privileges implementations are provided on the appendices.

Overall, effective database design and implementation require a careful and systematic approach, utilizing ER/EER diagrams for structured modeling, SQL for data manipulation, and advanced SQL features to augment functionality and manage data integrity.

# 12.Chapter 7: Conclusion

In conclusion, the development of a database management system for TT Holding effectively addresses the challenges of managing employee data. This new system enhances the processes of data collection, storage, and retrieval, significantly improving efficiency for the HR team.Following the methodology outlined by Rob (2018), the design transformed initial plans into a structured database. The careful organization of tables and relationships ensures data consistency and integrity, facilitating informed decision-making within the organization.

Moreover, the implementation of SQL functions, stored procedures, and triggers optimizes database performance and security. As highlighted by Navathe (2016), these enhancements are essential for maintaining effective data management. User engagement throughout the development process played a vital role in creating a system that aligns with the needs of the HR staff. Prioritizing user experience increases the likelihood of successful adoption across TT Holding.

Overall, this database management system serves as a critical tool for TT Holding, enabling improved workforce management and positioning the company for future growth and responsiveness to evolving business requirements.

# 13. References

1. Bourne, L. (2018). Project Management: A Systems Approach to Planning, Scheduling, and Control. 10th ed. Hoboken, NJ: John Wiley & Sons.

2. Coronel, C. and Morris, S. (2016). Database Systems: Complete Book. 2nd ed. Boston, MA: Cengage Learning.

3. Date, C. J. (2004). An Introduction to Database Systems. 8th ed. Boston, MA: Addison-Wesley.

4. Elmasri, R. and Navathe, S. B. (2015). Fundamentals of Database Systems. 7th ed. Boston, MA: Pearson.

5. Kendall, K. E. and Kendall, J. E. (2013). Systems Analysis and Design. 9th ed. Boston, MA: Pearson.

6. Kotter, J. P. (2007). Leading Change. Boston, MA: Harvard Business Review Press.

7. Navathe, S. B. and Elmasri, R. (2015). Fundamentals of Database Systems. 6th ed. Boston, MA: Addison-Wesley.

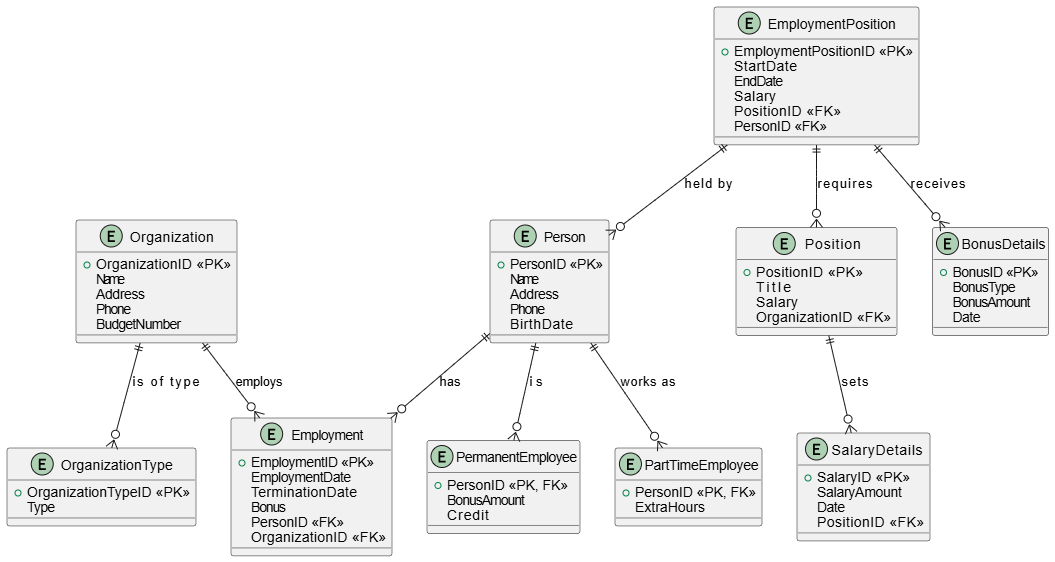
8. Navathe, S. B. (2016). Database Modeling and Design. 5th ed. Boston, MA: Morgan Kaufmann Publishers.

9. Rob, P. and Coronel, C. (2016). Database Systems: A Practical Approach to Design, Implementation, and Management. 5th ed. Boston, MA: Cengage Learning.

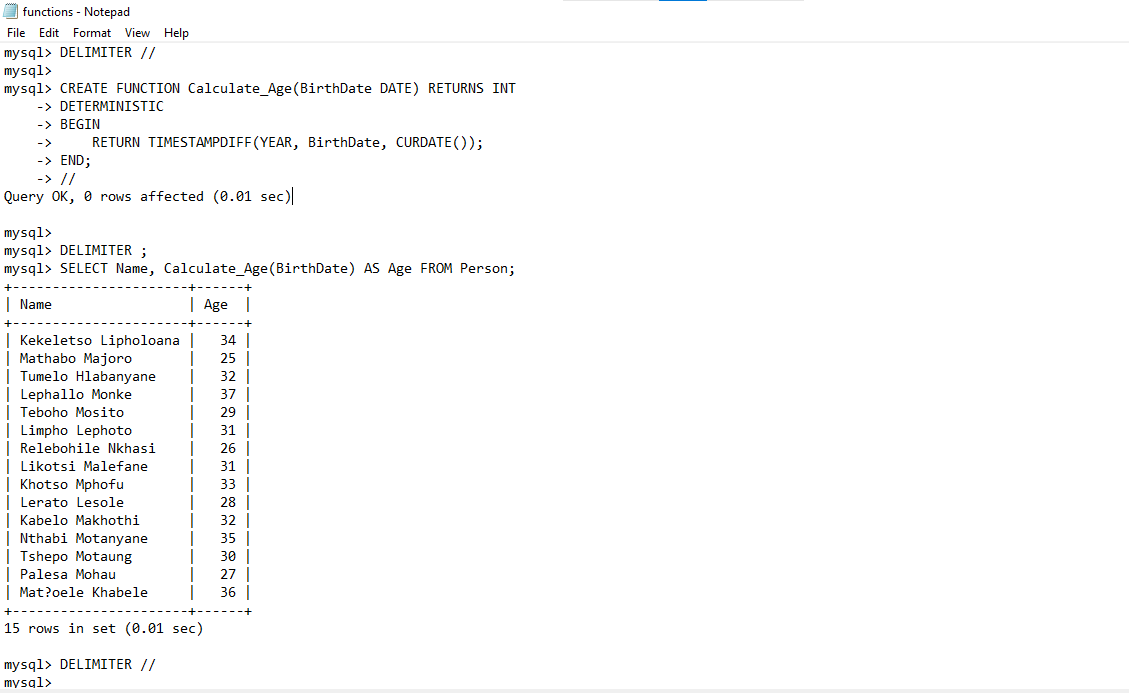
10. Sommerville, I. (2011). Software Engineering. 9th ed. Boston, MA: Addison-Wesley

# 16.Appendices of

1. Entity relationship diagram



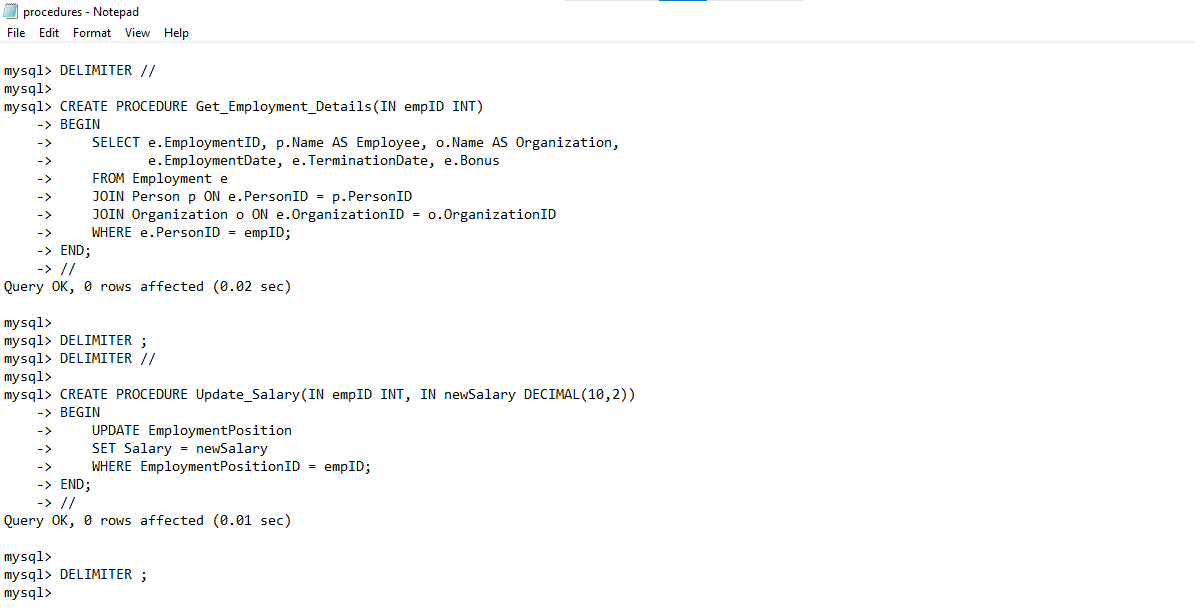
Functions



Triggers



Procedures



User creation and privileges

