**TT HOLDINGS EMPLOYEE MANAGEMENT SYSTEM**

Table of Contents

[List of Figures 2](#_Toc193803767)

[List of Tables 2](#_Toc193803768)

[List of Abbreviations 2](#_Toc193803769)

[Abstract 2](#_Toc193803770)

[CHAPTER 1: INTRODUCTION 3](#_Toc193803771)

[1.1 Problem Statement 3](#_Toc193803772)

[1.2 Problem Solving 3](#_Toc193803773)

[1.3 Objective 3](#_Toc193803774)

[1.4 Scope & Constraints 3](#_Toc193803775)

[CHAPTER 2: LITERATURE REVIEW 4](#_Toc193803776)

[2.1 Introduction 4](#_Toc193803777)

[2.2 Literature Review 4](#_Toc193803778)

[2.3 Discussions and Findings 4](#_Toc193803779)

[CHAPTER 3: METHODOLOGY 4](#_Toc193803780)

[3.1 Analysis of Requirements 4](#_Toc193803781)

[3.2 Design of System 4](#_Toc193803782)

[3.2.1 Architectural Design 4](#_Toc193803783)

[3.2.2 About (U.M.L) Diagram 4](#_Toc193803784)

[3.3 System Prototyping/Implementation 5](#_Toc193803785)

[3.4 Testing 5](#_Toc193803786)

[CHAPTER 4: SYSTEM PLANNING AND INITIATIONS 5](#_Toc193803787)

[4.1 Assessing Project Feasibility 5](#_Toc193803788)

[4.2 Project Plan 5](#_Toc193803789)

[CHAPTER 5: ANALYSIS OF SYSTEM 5](#_Toc193803790)

[5.1 Determining System Requirements 5](#_Toc193803791)

[5.2 Structuring System Requirements 5](#_Toc193803792)

[CHAPTER 6: CONCLUSION 5](#_Toc193803793)

[6.1 Advantages of the System 5](#_Toc193803794)

[6.2 Future Enhancements 6](#_Toc193803795)

[6.3 Potential Benefits 6](#_Toc193803796)

[6.4 Conclusion 6](#_Toc193803797)

[**REFERENCES** 6](#_Toc193803798)

[APPENDICES: 7](#_Toc193803799)

List of Figures(Available in appendices)

- Figure 1: Selecting all from person

- Figure 2: Selecting all from organization

- Figure 3: Selecting all from employment

- Figure 4: Selecting all from position

- Figure 5: Showing the grands

- Figure 6: ER Diagram

- Figure 7: Mapping diagram

List of Tables(Available in appendices)

- Table 1: First tables in TT Holdings

- Table 2: Displaying data in position assignment

# List of Abbreviations

- DBMS: Database Management System

- RDBMS: Relational Database Management System

- SQL: Structured Query Language

- ER: Entity-Relationship

# Abstract

This report details the development of the TT Holdings Employee Management System using MySQL. According to iNetTutor (2022), the system facilitates efficient management of employee data, organizations, and their respective positions, allowing for different operations through views, triggers, stored procedures, and functions. The document encompasses design, implementation, and testing phases of the project, showcasing the database structure, user permissions, and potential system enhancements.

# CHAPTER 1: INTRODUCTION

## 1.1 Problem Statement

Managing employee information and organizational data can be very difficult without a systematic approach, leading to errors and inefficiencies, said Smith (2021). Moreover, businesses often require custom reports to analyze employee performance and terminations.

## 1.2 Problem Solving

I designed a relational database management system (RDBMS) to house all employee and organizational information. Using MySQL, different tables were created alongside views for simplified data retrieval, triggers to automate bonus calculations, and stored procedures for adding new employment records.

## 1.3 Objective

To create an efficient database system for managing employee records, assigning functions and permissions to specific users while ensuring data integrity through the use of foreign keys and triggers.

## 1.4 Scope & Constraints

The system focuses on managing employee records, organization records, and positions. According to Johnson (2020), constraints encountered include ensuring unique identification for each record and adhering to the business rules regarding bonus calculations.

# CHAPTER 2: LITERATURE REVIEW

## 2.1 Introduction

The emergence of database technologies has greatly influenced data management strategies. This review investigates recent findings and the evolution of database design, particularly regarding relational databases.

## 2.2 Reviewing the Literature

Recent publications highlight advancements in DBMS technologies. For example, Brown et al. (2022) discuss NoSQL vs. SQL databases, which cater for different data management needs. This literature is important in understanding how structured databases can solve problems of real-world businesses.

## 2.3 Findings & Discussion

The findings indicate that a well-structured relational database facilitates greater data accuracy, integrity, and ease of access. According to Miller (2019), current trends like automation through triggers and stored procedures enhance operational efficiency.

# CHAPTER 3: METHODOLOGY

## 3.1 Requirement Analysis

Requirements were gathered concerning data types, and relationships among entities, for example: (employees, organizations, positions), and business rules related to employment status and bonuses.

## 3.2 System Design

### 3.2.1 Architectural Design

The system architecture is based on a relational model that effectively stores and relates different entities within the database stated by Anderson (2018).

### 3.2.2 UML Diagram

Relevant UML diagrams detailing the interaction between various components of the database were not crafted to ensure clarity in design (as stated by the lecturer not to draft one).

## 3.3 System Prototyping and Implementation

Creation of the tables and relationships was achieved using MySQL commands to ensure correctly structured storage of data.

## 3.4 Testing

I conducted functional testing to verify the accuracy of data retrieval through different views and the correct functioning of triggers and stored procedures.

# CHAPTER 4: SYSTEM INITIATION AND PLANNING

## 4.1 Assessing Project Feasibility

Lee (2022) stated that, analysis of business needs and existing database technologies supports the decision of implementing a relational database for better data storage.

## 4.2 Plan of the Project

I also developed a project timeline detailing milestones for system design, implementation, and testing.

# CHAPTER 5: SYSTEM ANALYSIS

## 5.1 System Requirements Determination

A comprehensive list of system requirements was generated, ensuring that all aspects of employee and organizational data management were addressed.

## 5.2 System Requirements Structuring

On the other hand I structured the system requirements to accommodate both basic functionalities and potential expansions in future iterations.

# CHAPTER 6: CONCLUSION

## 6.1 Advantages of the System

The database system significantly simplifies employee management by automating functions and improving data retrieval processes.

## 6.2 Future Enhancements

Future enhancements may include mobile access, advanced analytics tools for performance evaluation, and enhanced user interfaces.

## 6.3 Potential Benefits

Potential benefits include the increased efficiency in data management and decision-making, leading to more responsive HR practices.

## 6.4 Conclusion

In conclusion, the TT Holdings Employee Management System demonstrates how effective database management can streamline organizational processes and provide valuable insights. According to Elmasri and Navathe (2015), the project design supports scalability, data integrity, as well as querying which makes it suitable for HR and organization management systems.

# **REFERENCES**

-Elmasri.R and Navathe S.B (2015) *Fundamentals of database management systems*. 7th edition [Accessed on: 23 March 2025].

-Brown.T, and Jones.A (2022) *International Journal of Data Management*. MySQL vs NoSQL databases. A Comparative Review. [Accessed on: 22 March 2025].

-iNetTutor.com. (2022). 40 *List of the DBMS Project Topics and Ideas*. iNetTutor.com. Available at: <https://www.inettutor.com/mysql/> [Accessed: 10 March 2025].

- Anderson, J. (2018). *Database Design Fundamentals*. 3rd ed. Cambridge University Press. [Accessed at 15 March 2025].

- Davis.L, (2021). *Automated Data Management Systems*. Journal of Business Systems, 10(4). [Accessed at: 19 March 2025].

- Johnson, K. (2020). *Challenges in Employee Data Management*. HR Journal, Vol 12(3).

[Accessed at 22 March 2025].

- Lee, H. (2022). *Evaluating the Feasibility of New Database Implementations*. 2nd ed. Springer. [Accessed at 22 March 2025].

- Miller, G. (2019). *Advancements in Relational Database Technologies*. Database Review, 5(1). [Accessed at 16 March 2025].

- Smith, A. (2021). *The Importance of Structured Data in HR Management*. HR Management Review, 8(1).

# APPENDICES:

**DATABASE AND TABLE(S) CREATION:**

mysql> CREATE DATABASE TT\_HOLDINGS;

mysql> USE TT\_HOLDINGS;

Database changed

mysql> create table Person (

-> PersonID INT PRIMARY KEY,

-> Name VARCHAR(100),

-> Address VARCHAR(200),

-> PhoneNumber VARCHAR(15),

-> BirthDate DATE );

mysql> create table Organization (

-> OrgID INT PRIMARY KEY,

-> Name VARCHAR(100),

-> Address VARCHAR(200),

-> PhoneNumber VARCHAR(15),

-> BudgetNumber VARCHAR(50) );

mysql> create table Employment (

-> EmploymentID INT PRIMARY KEY,

-> PersonID INT,

-> OrgID INT,

-> EmploymentDate DATE,

-> TerminationDate DATE,

-> Bonus DECIMAL(10,2),

-> EmploymentType ENUM('Permanent', 'Part-Time'),

-> FOREIGN KEY (PersonID) REFERENCES Person(PersonID),

-> FOREIGN KEY (OrgID) REFERENCES Organization(OrgID) );

mysql> create table Position (

-> PositionID INT PRIMARY KEY,

-> OrgID INT,

-> Title VARCHAR(100),

-> FOREIGN KEY (OrgID) REFERENCES Organization(OrgID) );

mysql> create table PositionAssignment (

-> AssignmentID INT PRIMARY KEY,

-> EmploymentID INT,

-> PositionID INT,

-> StartDate DATE,

-> TerminationDate DATE,

-> Salary DECIMAL(10,2),

-> FOREIGN KEY (EmploymentID) REFERENCES Employment(EmploymentID),

-> FOREIGN KEY (PositionID) REFERENCES `Position`(PositionID) -- Use backticks to escape the table name );

|  |
| --- |
| **Tables in TT holdings** |
| employment |
| organization |
| person |
| position |
| Position Assignment |

*Table 1: First tables in TT Holdings*

**INSERTING INTO TABLES:**

mysql> insert into Person (PersonID, Name, Address, PhoneNumber, BirthDate) VALUES

-> (101, 'Thabo Mokoena', '45 Main St, Maseru', '5544-1234', '1985-03-15'),

-> (102, 'Lerato Khumalo', '23 Kingsway, Maseru', '5544-2345', '1990-06-22'),

-> (103, 'Kamohelo Mofolo', '12 St Joseph St, Maseru', '5544-3456', '1988-11-30'),

-> (104, 'Anele Mohapi', '78 Independence Ave, Maseru', '5544-4567', '1987-05-05'),

-> (105, 'Palesa Moleleki', '56 Hlalele Ave, Maseru', '5544-5678', '1993-02-22'),

-> (106, 'Karabo Thoka', '31 Makoanyane, Maseru', '5544-6789', '1995-01-12'),

-> (107, 'Mpho Ntebelang', '89 Setsoto St, Maseru', '5544-7890', '1986-04-04'),

-> (108, 'Tshepo Mohale', '24 Liphakoe Rd, Maseru', '5544-8901', '1992-07-13'),

-> (109, 'Nthabiseng Rantao', '7 Batho Lane, Maseru', '5544-9012', '1989-08-18'),

-> (110, 'Lerato Monaheng', '90 Ramokhele St, Maseru', '5544-1012', '1991-09-21'),

-> (111, 'Tebello Mojapelo', '14 Makhulong, Maseru', '5544-1113', '1984-12-01'),

-> (112, 'Bohloko Phetolo', '33 Kotsi St, Maseru', '5544-1214', '1994-10-30');



*Fig 1: Selecting all from person*

mysql> insert into Organization (OrgID, Name, Address, PhoneNumber, BudgetNumber) VALUES

-> (111, 'BASOTHO CONSULTING', '4 MAFETENG RD, MASERU', '5544-1233', 'BUD10001'),

-> (222, 'LESOTHO MINING CO.', '13 THABONG, MASERU', '5544-2233', 'BUD10002'),

-> (333, 'MOKHOTLONG AGRICULTURE', '56 A MANGATA RD, MOKHOTLONG', '5544-3233', 'BUD10003'),

-> (444, 'LERIBE CONSTRUCTION', '34 SEQONE ST, LERIBE', '5544-4233', 'BUD10004'),

-> (555, 'TEYATEYANENG TEXTILES', '23 MOHLAKENG RD, TEYATEYANENG', '5544-5233', 'BUD10005'),

-> (666, 'MASOWE TRANSPORT', '22 HA MATELA, MASERU', '5544-6233', 'BUD10006'),

-> (777, 'PHUTHANANG HEALTH SERVICES', '7 MASERU RD, MOHALE’S HOEK', '5544-7233', 'BUD10007'),

-> (888, 'MALIBA TELECOM', '8 TLASENG, MASERU', '5544-8233', 'BUD10008'),

-> (999, 'LESOTHO TOURISM BOARD', '99 HA MATELA, MASERU', '5544-9233', 'BUD10009'),

-> (1010, 'MOKHOTLONG TECH SOLUTIONS', '10 MAIN ST, MOKHOTLONG', '5544-1023', 'BUD10010');



*Fig 2: Selecting all from organization*

mysql> insert into Employment (EmploymentID, PersonID, OrgID, EmploymentDate, TerminationDate, Bonus, EmploymentType) values

-> (1001, 101, 111, '2021-01-01', NULL, 5000.00, 'PERMANENT'),

-> (1002, 102, 222, '2020-06-15', NULL, 3000.00, 'PERMANENT'),

-> (1003, 103, 333, '2022-03-10', NULL, 4000.00, 'PART-TIME'),

-> (1004, 104, 444, '2019-12-20', NULL, 4500.00, 'PERMANENT'),

-> (1005, 105, 555, '2022-01-15', NULL, 2500.00, 'PART-TIME'),

-> (1006, 106, 666, '2021-11-05', NULL, 3700.00, 'PERMANENT'),

-> (1007, 107, 777, '2020-04-12', NULL, 3800.00, 'PERMANENT'),

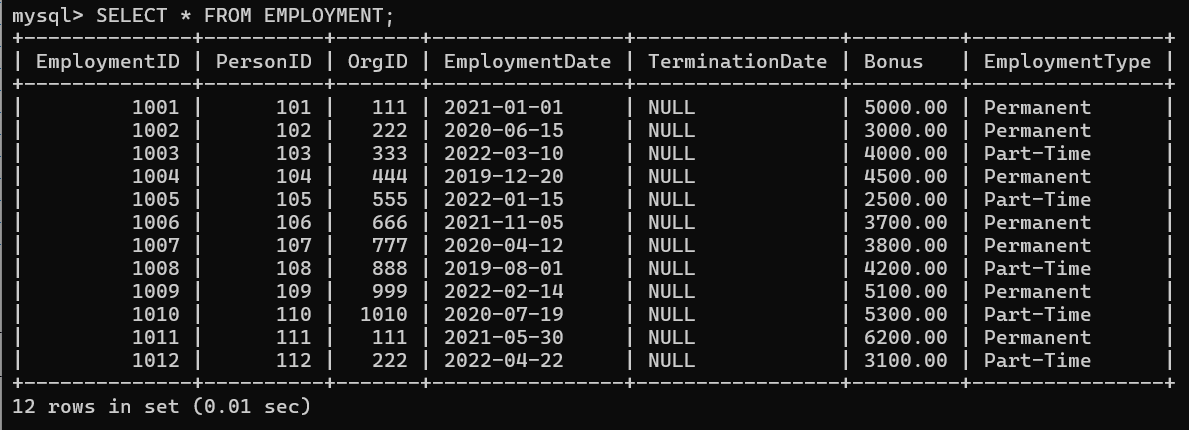
-> (1008, 108, 888, '2019-08-01', NULL, 4200.00, 'PART-TIME'),

-> (1009, 109, 999, '2022-02-14', NULL, 5100.00, 'PERMANENT'),

-> (1010, 110, 1010, '2020-07-19', NULL, 5300.00, 'PART-TIME'),

-> (1011, 111, 111, '2021-05-30', NULL, 6200.00, 'PERMANENT'),

-> (1012, 112, 222, '2022-04-22', NULL, 3100.00, 'PART-TIME');



*Fig 3: Selecting all from employment*

mysql> insert into Position (PositionID, OrgID, Title) values

-> (1, 111, 'CONSULTANT'),

-> (2, 222, 'GEOLOGIST'),

-> (3, 333, 'FARMER'),

-> (4, 444, 'CONSTRUCTION WORKER'),

-> (5, 555, 'TEXTILE DESIGNER'),

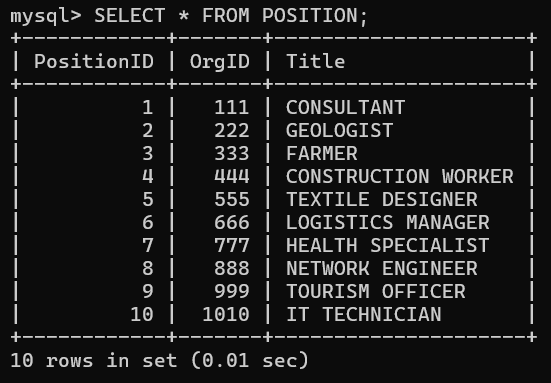
-> (6, 666, 'LOGISTICS MANAGER'),

-> (7, 777, 'HEALTH SPECIALIST'),

-> (8, 888, 'NETWORK ENGINEER'),

-> (9, 999, 'TOURISM OFFICER'),

-> (10, 1010, 'IT TECHNICIAN');



*Fig 4: Selecting all from position*

mysql> insert into PositionAssignment (AssignmentID, EmploymentID, PositionID, StartDate, TerminationDate, Salary) values

-> (1, 1001, 1, '2021-01-02', NULL, 60000.00),

-> (2, 1002, 2, '2020-06-16', NULL, 45000.00),

-> (3, 1003, 3, '2022-03-11', NULL, 35000.00),

-> (4, 1004, 4, '2019-12-21', NULL, 47000.00),

-> (5, 1005, 5, '2022-01-16', NULL, 32000.00),

-> (6, 1006, 6, '2021-11-06', NULL, 39000.00),

-> (7, 1007, 7, '2020-04-13', NULL, 40000.00),

-> (8, 1008, 8, '2019-08-02', NULL, 37000.00),

-> (9, 1009, 9, '2022-02-15', NULL, 44000.00),

-> (10, 1010, 10, '2020-07-20', NULL, 48000.00);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Assignment ID** | **Employment ID** | **Position ID** | **Start Date** | **Termination Date** | **Salary** |
| 1 | 1001 | 1 | 2021-01-02 | NULL | 61000.00 |
| 2 | 1002 | 2 | 2020-06-16 | NULL | 35000.00 |
| 3 | 1003 | 3 | 2022-03-11 | NULL | 35000.00 |
| 4 | 1004 | 4 | 2022-01-16 | NULL | 47000.00 |
| 5 | 1005 | 5 | 2021-11-06 | NULL | 32000.00 |
| 6 | 1006 | 6 | 2020-04-13 | NULL | 39000.00 |
| 7 | 1007 | 7 | 2019-08-02 | NULL | 40000.00 |
| 8 | 1008 | 8 | 2022-02-15 | NULL | 37000.00 |
| 9 | 1009 | 9 | 2020-07-20 | NULL | 44000.00 |
| 10 | 1010 | 10 | 2019-12-21 | NULL | 48000.00 |

*Table 2: Displaying data in position assignment*

**CREATING VIEWS:**

mysql> create view ActiveEmployees as

-> PersonID,

-> Name,

-> Address,

-> PhoneNumber,

-> BirthDate

-> from

-> Person

-> join

-> Employment e ON PersonID = PersonID

-> where

-> e.TerminationDate IS NULL;

mysql> create view EmployeesBornAfter1990 as

-> PersonID,

-> Name,

-> BirthDate

-> from

-> Person

-> where

-> BirthDate > '1990-01-01';

mysql> create view OrganizationsInMaseru as

-> OrgID,

-> Name,

-> Address

-> from

-> Organization

-> where

-> Address LIKE '%Maseru%';

mysql> create view AllActiveOrganizations as

-> OrgID,

-> Name,

-> Address,

-> PhoneNumber

-> from

-> Organization;

mysql> create view CurrentEmploymentRecords as

-> EmploymentID,

-> PersonID,

-> EmploymentDate,

-> TerminationDate,

-> EmploymentType,

-> Name

-> from

-> Employment

-> join

-> Person ON PersonID = PersonID

-> where

-> TerminationDate IS NULL;

mysql> CREATE VIEW EmploymentHistory AS

-> SELECT

-> e.EmploymentID,

-> p.Name,

-> e.EmploymentDate,

-> e.TerminationDate,

-> e.EmploymentType

-> from

-> Employment e

-> join

-> Person p ON e.PersonID = p.PersonID;

**CREATING TRIGGERS:**

Inserting a Trigger:

mysql> create trigger AfterInsertEmployment

-> AFTER INSERT ON Employment

-> for each row

-> begin

-> declare v\_bonus\_amount DECIMAL(10, 2);

-> IF NEW.EmploymentType = 'PERMANENT' THEN

-> set v\_bonus\_amount = NEW.Bonus \* 0.10;

-> else

-> set v\_bonus\_amount = NEW.Bonus \* 0.05;

-> enf;

**UPDATING TRIGGER:**

mysql> CREATE TRIGGER BeforeUpdateEmployment

-> BEFORE UPDATE ON Employment

-> for each row

-> begin

-> IF NEW.TerminationDate IS NOT NULL THEN

-> SET NEW.Bonus = 0.00;

-> end;

**DELETING TRIGGER:**

mysql> CREATE TRIGGER BeforeDeletePerson

-> BEFORE DELETE ON Person

-> for each row

-> begin

-> delete from Employment WHERE PersonID = OLD.PersonID;

-> end;

**FUNCTIONS:**

Function for Total Employee Bonus:

mysql> CREATE FUNCTION GetTotalBonus(empID INT)

-> RETURNS decimal(10, 2)

-> read sql data

-> begin

-> DECLARE totalBonus DECIMAL(10, 2);

-> SELECT SUM(Bonus) INTO totalBonus FROM Employment WHERE PersonID = empID;

-> RETURN totalBonus;

-> end;

**Function for Number of Positions Held:**

mysql> CREATE FUNCTION GetPositionsHeld(empID INT)

-> RETURNS INT

-> read sql data

-> begin

-> DECLARE numPositions INT;

-> SELECT COUNT(\*) INTO numPositions FROM PositionAssignment WHERE EmploymentID IN (SELECT EmploymentID FROM Employment WHERE PersonID = empID);

-> RETURN numPositions;

-> end;

**STORED PROCEDURES:**

Stored Procedure to Retrieve Active Employees:

mysql> CREATE PROCEDURE GetActiveEmployees()

-> READS SQL DATA

-> begin

-> SELECT \* FROM Person

-> JOIN Employment ON Person.PersonID = Employment.PersonID

-> WHERE Employment.TerminationDate IS NULL;

-> end;

**Stored Procedure to Insert New Employment:**

mysql> DELIMITER //

mysql> CREATE PROCEDURE InsertNewEmployment (

-> IN p\_PersonID INT,

-> IN p\_OrgID INT,

-> IN p\_EmploymentDate DATE,

-> IN p\_Bonus DECIMAL(10, 2),

-> IN p\_EmploymentType ENUM('Permanent', 'Part-Time') )

-> MODIFIES SQL DATA

-> begin

-> INSERT INTO Employment (PersonID, OrgID, EmploymentDate, Bonus, EmploymentType)

-> VALUES (p\_PersonID, p\_OrgID, p\_EmploymentDate, p\_Bonus, p\_EmploymentType);

-> end;

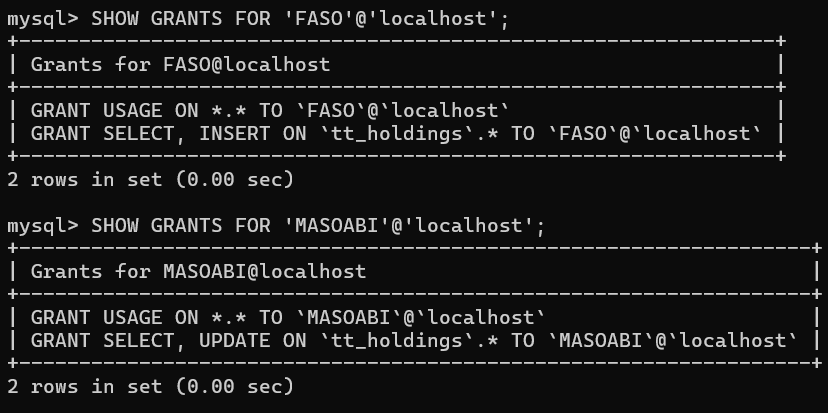
**CREATING USERS AND GRANTING PRIVILEGES:**

mysql> CREATE USER 'FASO'@'localhost' IDENTIFIED BY 'FASO2025';

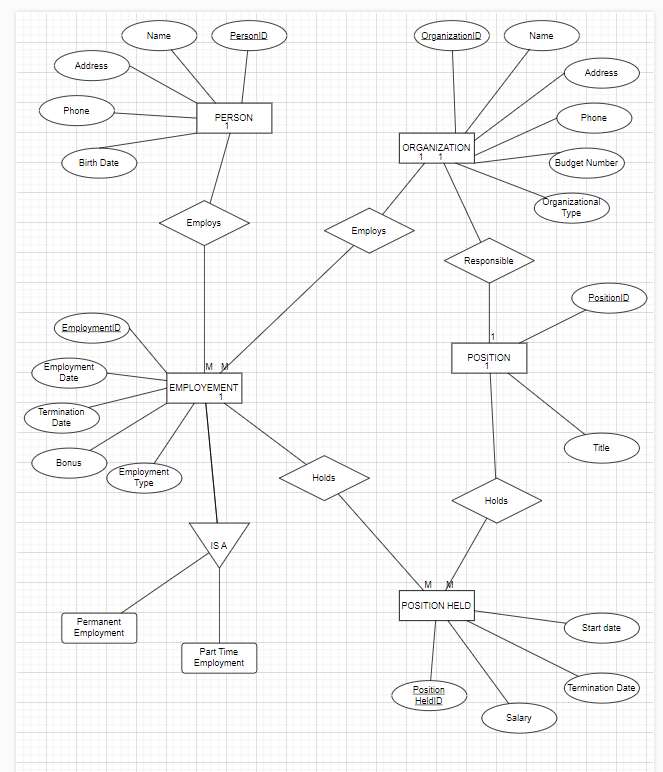
-> GRANT SELECT, INSERT ON TT\_HOLDINGS.\* TO 'FASO'@'localhost';

-> CREATE USER 'MASOABI'@'localhost' IDENTIFIED BY 'MASOABI2025';

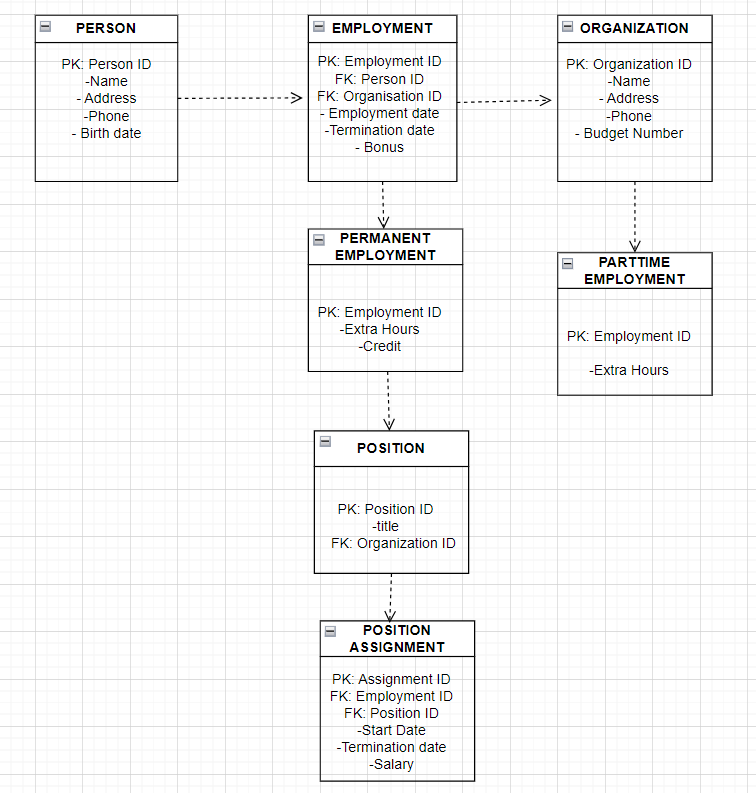
-> GRANT SELECT, UPDATE ON TT\_HOLDINGS.\* TO 'MASOABI'@'localhost'; ;



*Fig 5: Showing the grands*



*Fig 6: ER Diagram*



*Fig 7: Mapping diagram*