****

**Title Page:**

**TT-Holdings Database Management System**

**Author:** Nonyana Mohale

**ID Number:**901017208

**Institution:** Limkokwing University of Technology Lesotho

**Department:** Faculty of Information & Communication Technology

**Course Name:** BSc in Information Technology

**Program:** BIDB2212 . Database Systems

**Indivudual Assignment**

**Date:** 28/03/2025

**Lecturer:** Mr T Makheka

­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Table of Contents:

1. **Abstract** ...................................................................................................... 2
2. **List of Figures** ........................................................................................................ 3
3. **List of Abbreviations** .............................................................................................. 4

**Chapter 1: Introduction**

1.1 ProblemStatement........................................................................................... 5  
1.2 ProposedSolutions............................................................................................ 6  
1.3 Objectives......................................................................................................... 7  
1.4 Scope and Constraints...................................................................................... 8

**Chapter 2: Literature Review**

2.1 Evolution of HR Database Systems.................................................................. 9  
2.2 Critical Review of Existing Solutions................................................................ 10  
2.3 Research Gaps and Project Alignment............................................................ 11

**Chapter 3: Methodology**

3.1 Requirement Analysis .................................................................................... 12  
3.2 System Design  
 3.2.1 Architectural Design ............................................................................... 13  
 3.2.2 UML Diagrams ........................................................................................ 14  
3.3 System Implementation ................................................................................ 15  
3.4 Testing Procedures ........................................................................................ 16

**Chapter 4: System Initiation and Planning**

4.1 Feasibility Study ............................................................................................ 17  
4.2 Project Plan and Timeline.............................................................................. 18

**Chapter 5: System Analysis**

5.1 Functional Requirements .............................................................................. 19  
5.2 Non-Functional Requirements ...................................................................... 20

**Chapter 6: Conclusion**

6.1 Key Advantages ............................................................................................. 21  
6.2 Future Enhancement Roadmap ..................................................................... 22  
6.3 Broader Implications ..................................................................................... 23  
6.4 Final Reflections ............................................................................................ 24

**Reference Sections**

1. **References** ............................................................................................... 25
2. **Appendices**  
    Appendix A: Complete SQL Scripts ....................................................... 26  
    Appendix B: Sample HR Forms ............................................................. 27  
    Appendix C: Test Datasets .................................................................... 28

6. Abstract:

This report describes how we designed and developed the TT\_Holdings Employment Database Management System using MySQL. Monitoring employees, employers, occupations, and employment histories is made easier by the system. To follow business rules, it also makes use of triggers, stored procedures, and views. The database automatically calculates bonuses, produces well-structured reports, and checks the data's accuracy.

1. **List of Figures:**

Figure1.1 ***ER Diagram:***

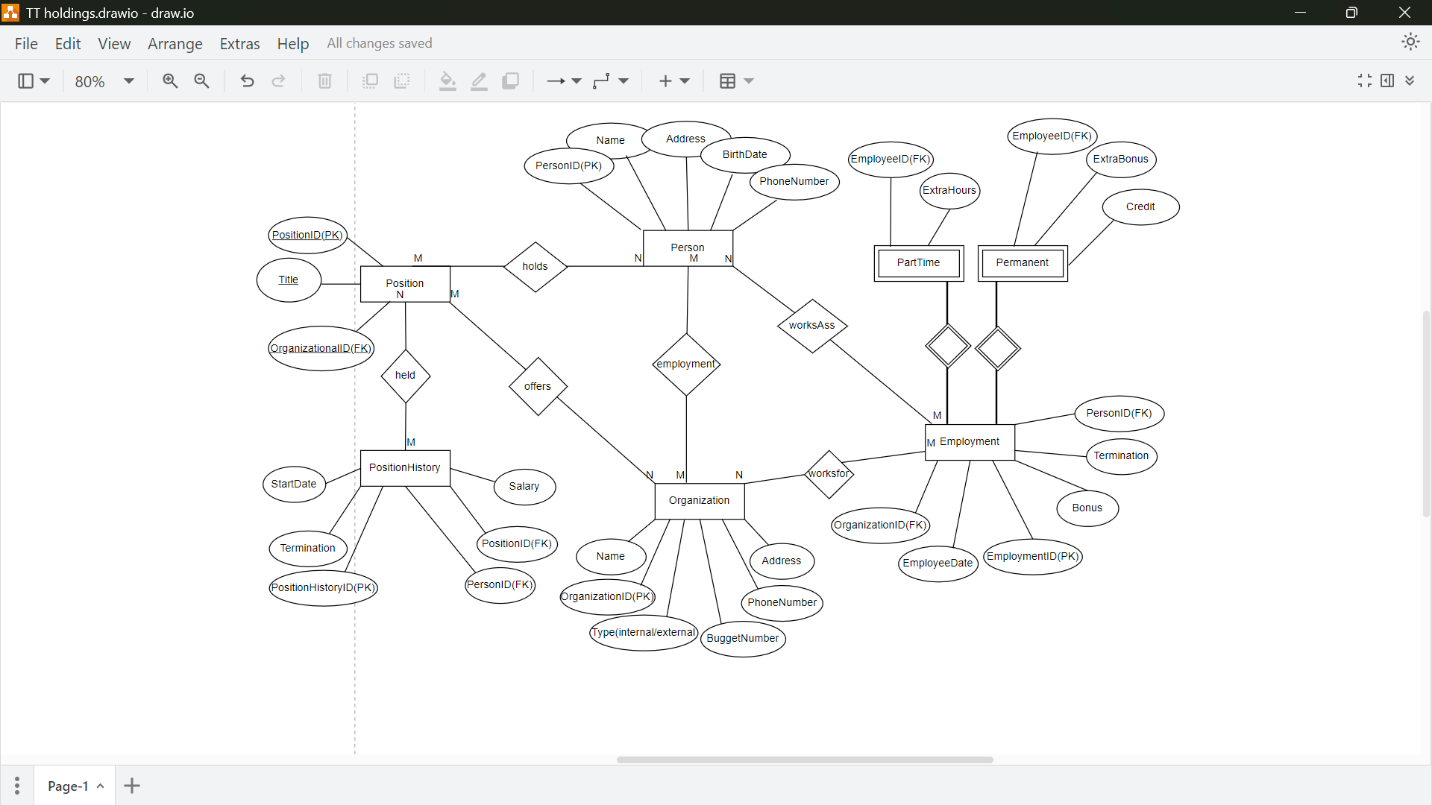
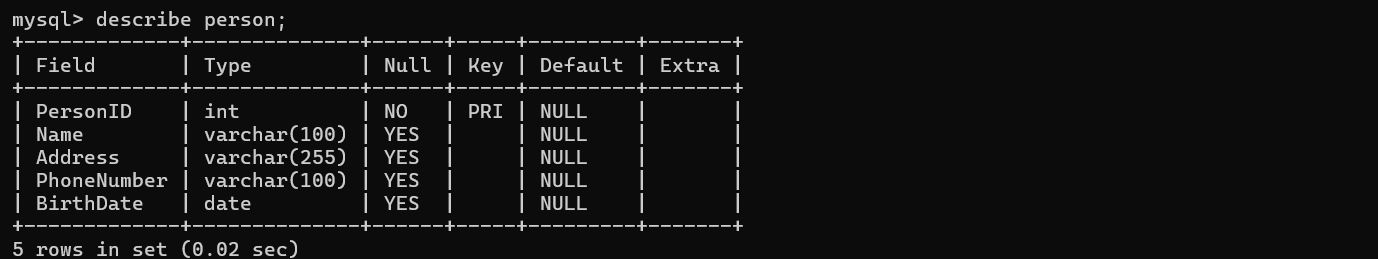


Figure1.2 ***MAPPING:***

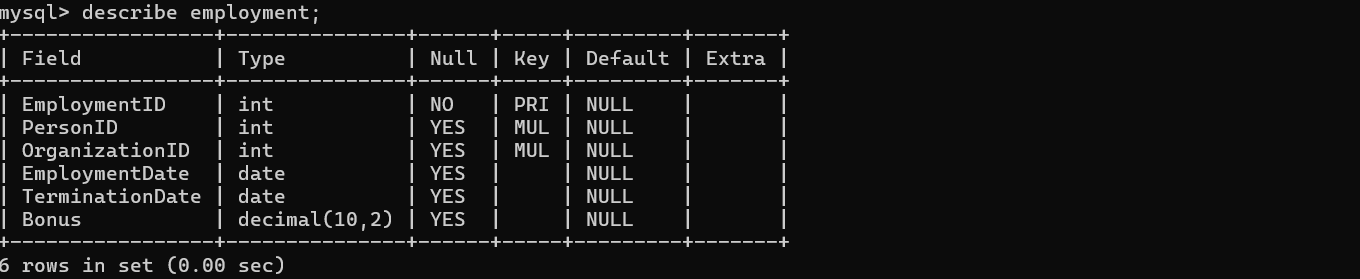
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Table** | **Primary Key** | **Foreign Key** | **Attributes** |
| Person | Person | PersonID | None | Name,Address,PhoneNumber  BirthDate |
| Organization | Organization | OrganizationID | None | Name,Address,PhoneNumber  BudgetNumber,Type |
| Employment | Employment | EmploymentID | PersonID, OrganizationID | EmploymentDate,  TerminationDate,Bonus |
| Position | Position | PositionID | OrganizationID | Title |
| PositionHistory | PositiobHistory | PositionHistoryID | PersonID, PositionID | StartDate,TerminationDate,  Salary |
| PermanentEmployment | PermanentEmployment | EmploymentID | EmploymentID | ExtraBonus, Credit |
| ParttimeEmployment | ParttimeEmployment | EmploymentID | EmploymentID | ExtraHours |

***Figure2.1 Database (and tables) creation & Insertion:***

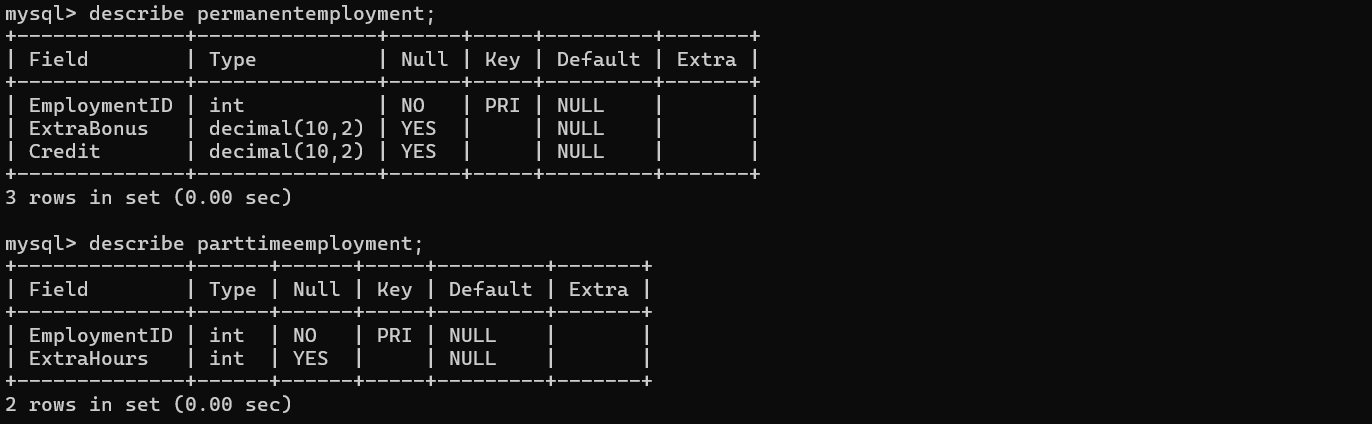
******

**Figure 3.1** Describe :******

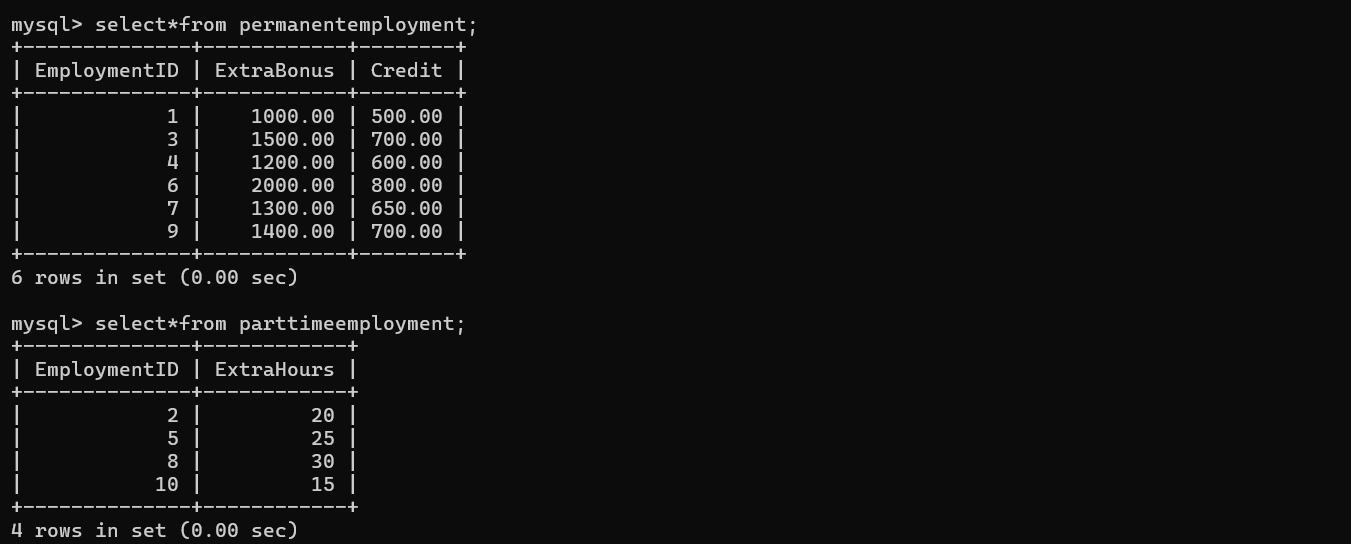
**Figure 3.2** Data insertion:******

**Figure 4.1** Describe***:***

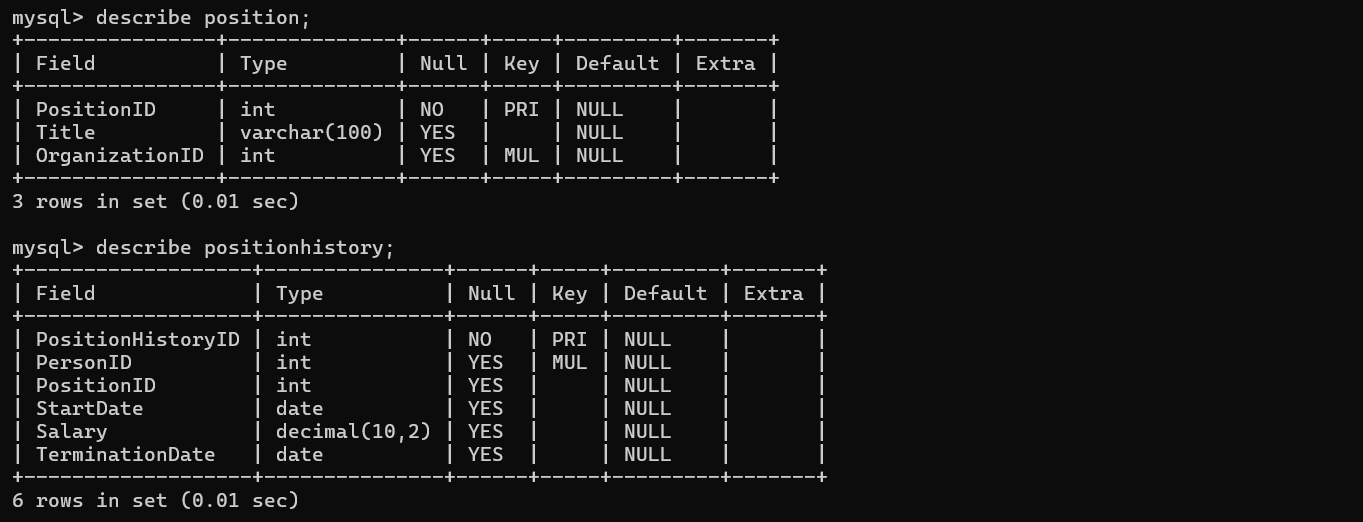
**Figure 4.2** Data insertion:******

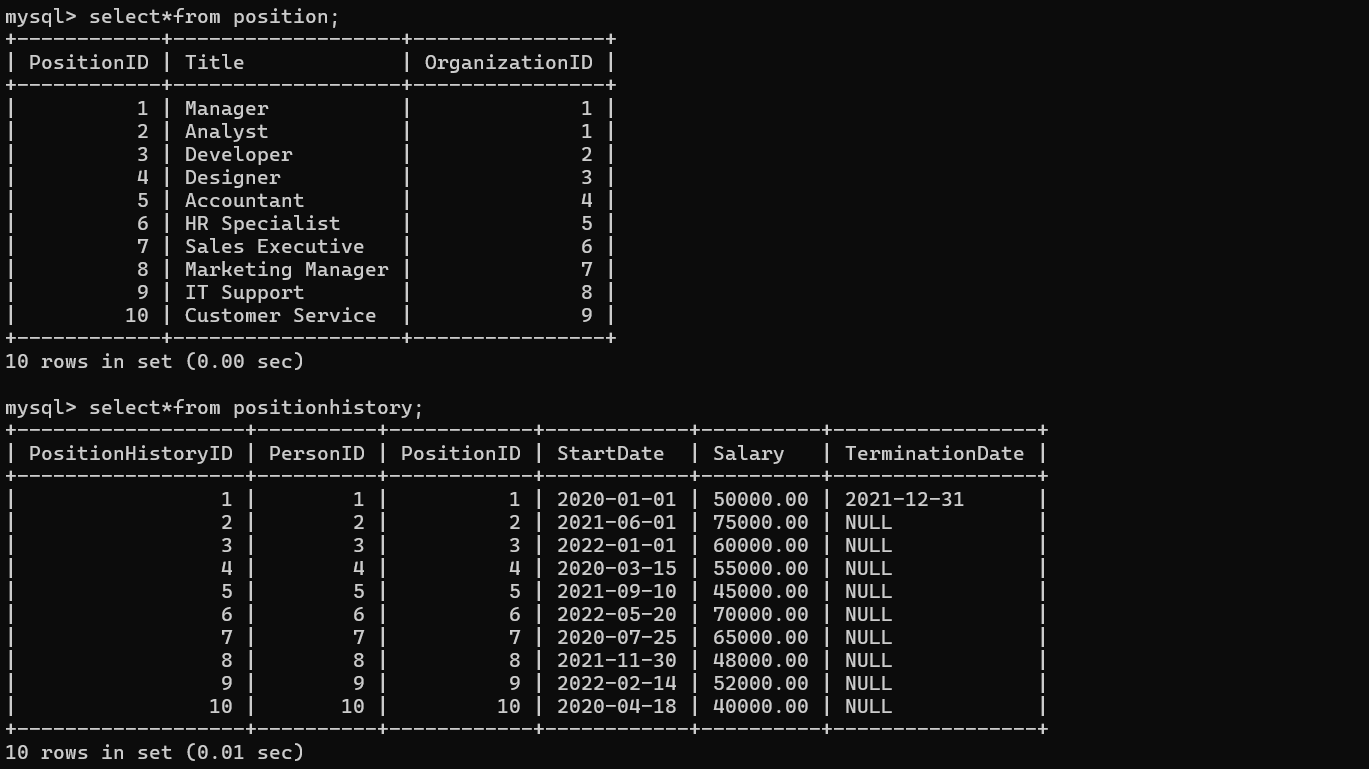
**Figure 5.1.** Describe:******

**Figure 5.2** Data insertion:

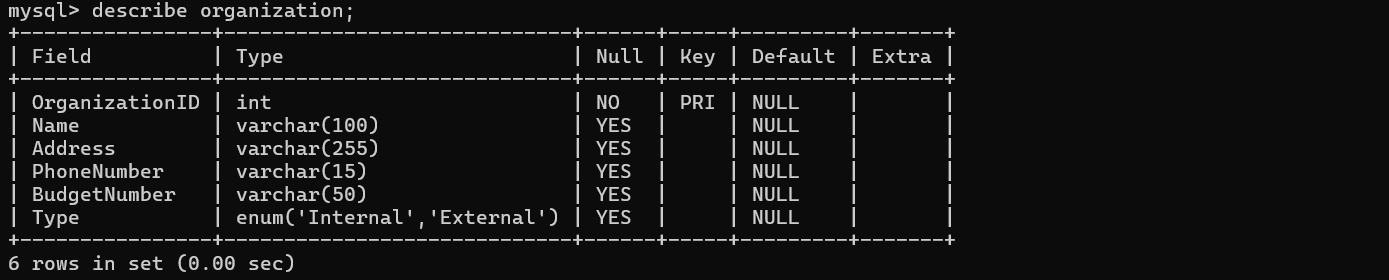
******

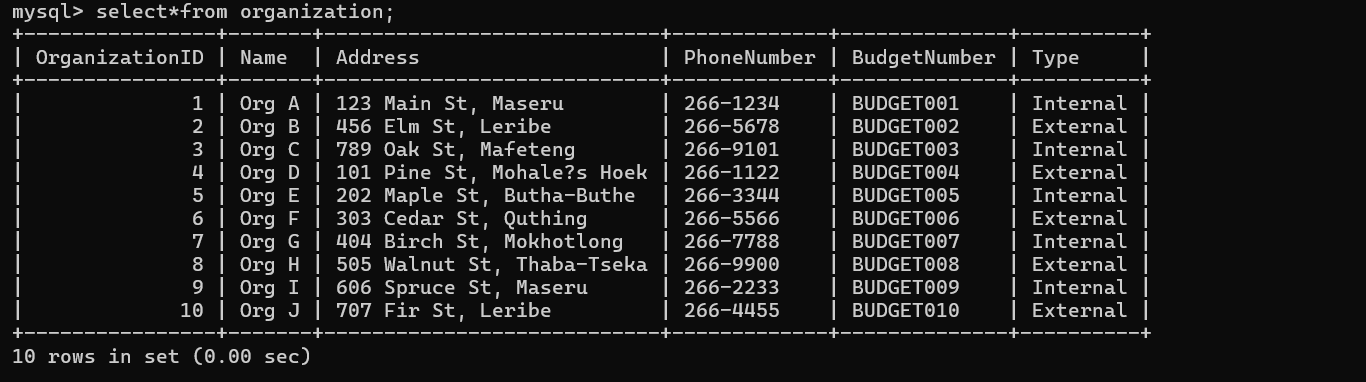
**Figure 6.1** Describe***:***

******

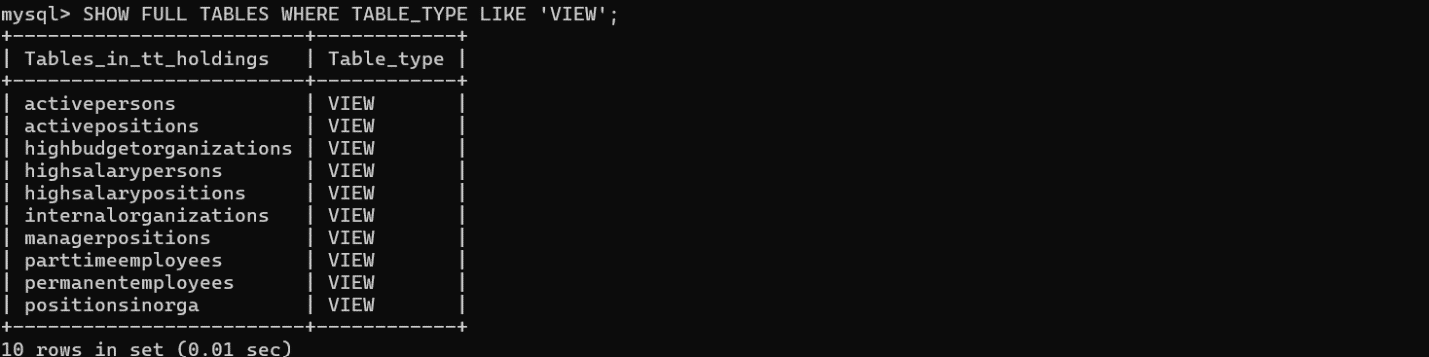
**Figure 6.2** Data insertion:******

**Figure 7.1** Describe:

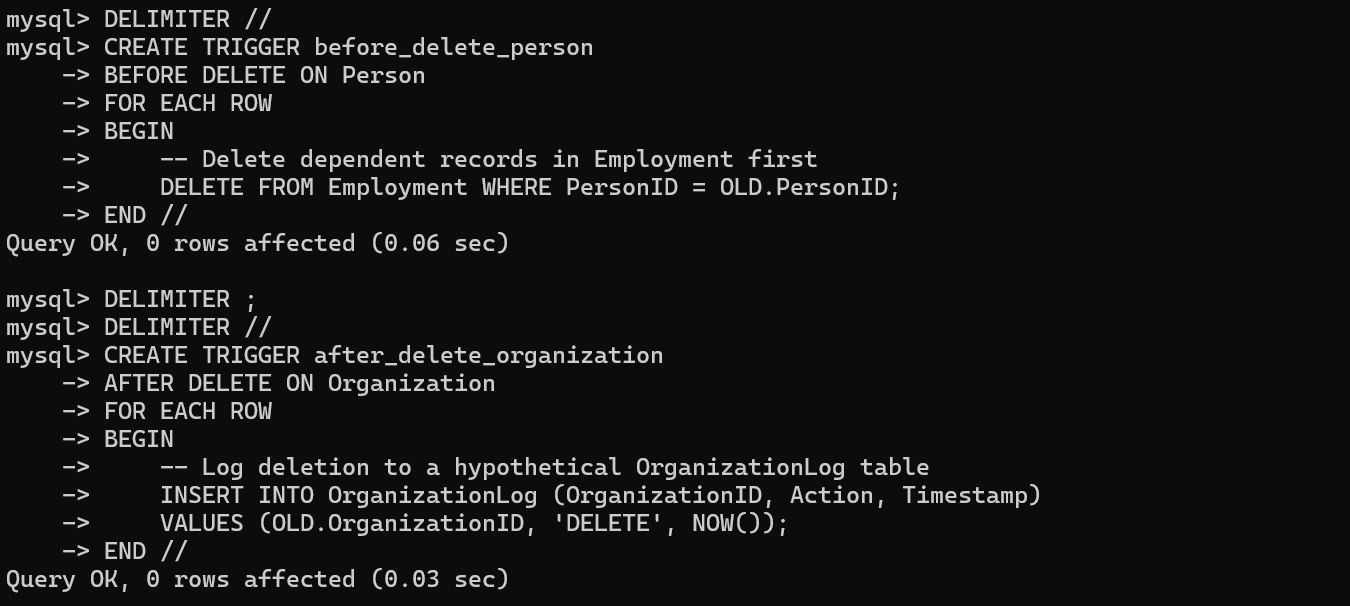
******

**Figure 7.2** Data insertion:******

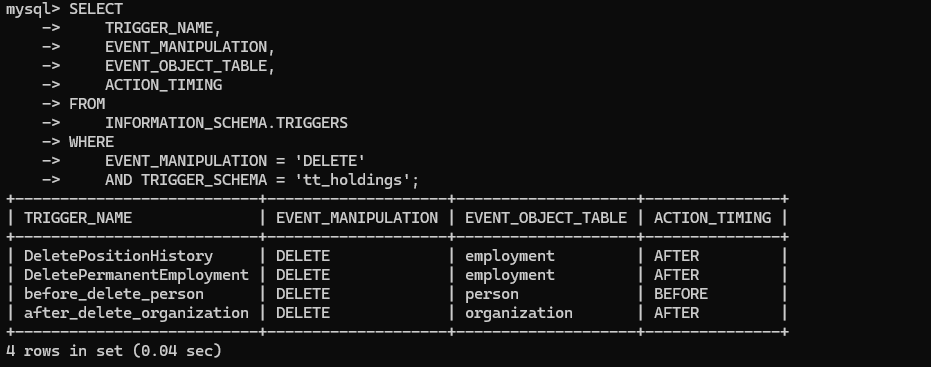
**Figure: 8.1 Views Created**

****

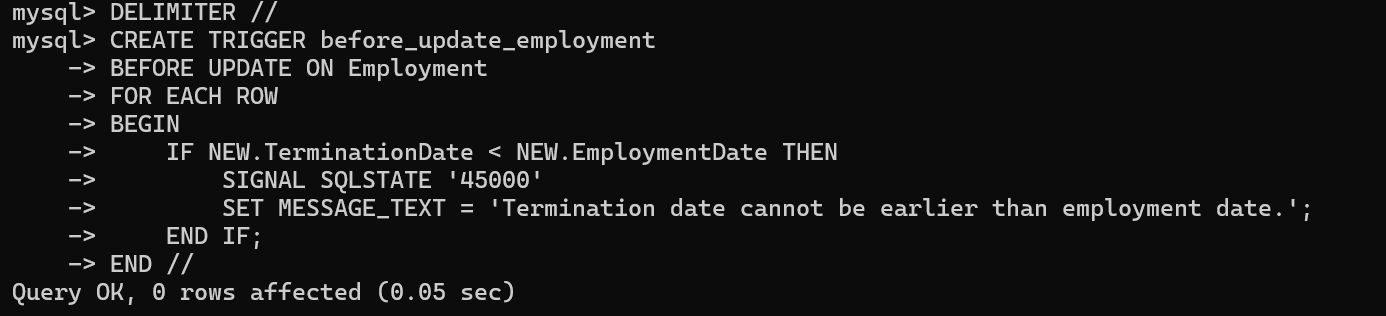
**Figure: 9.1Created Two Delete Triggers**

****

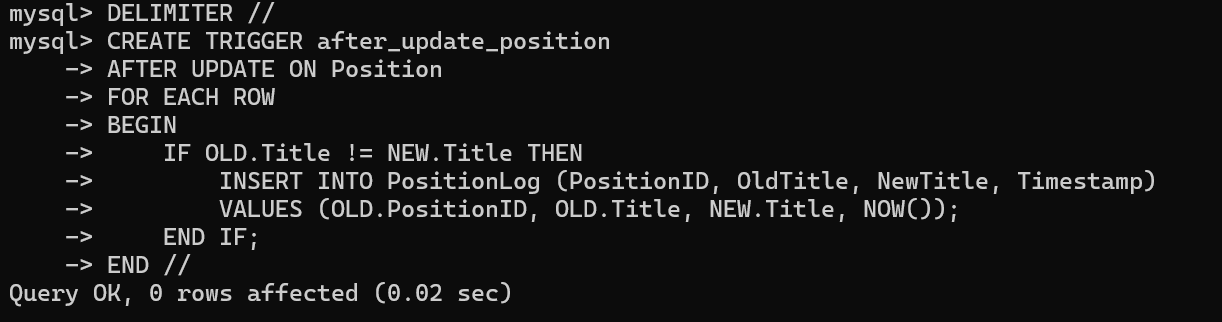
**Figure:9.1.1 Show Two Delete Triggers**

****

**Figure: 9.2.1 1st Update Triggers**



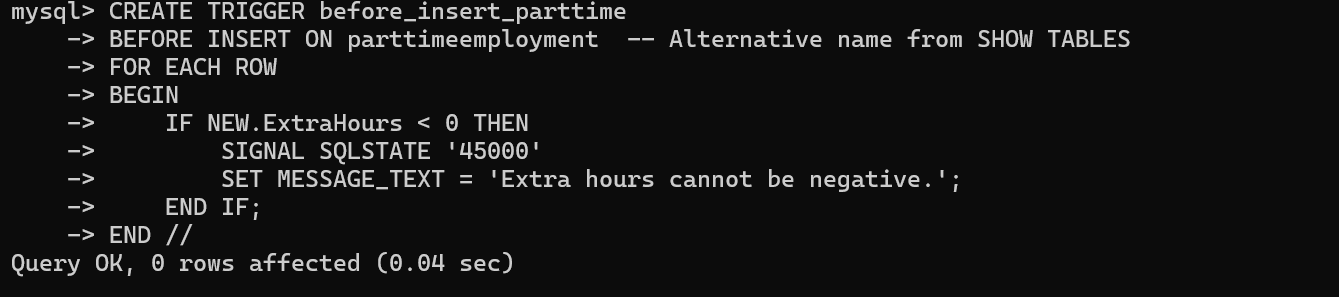
**Figure: 9.2.2 2nd Update Triggers**

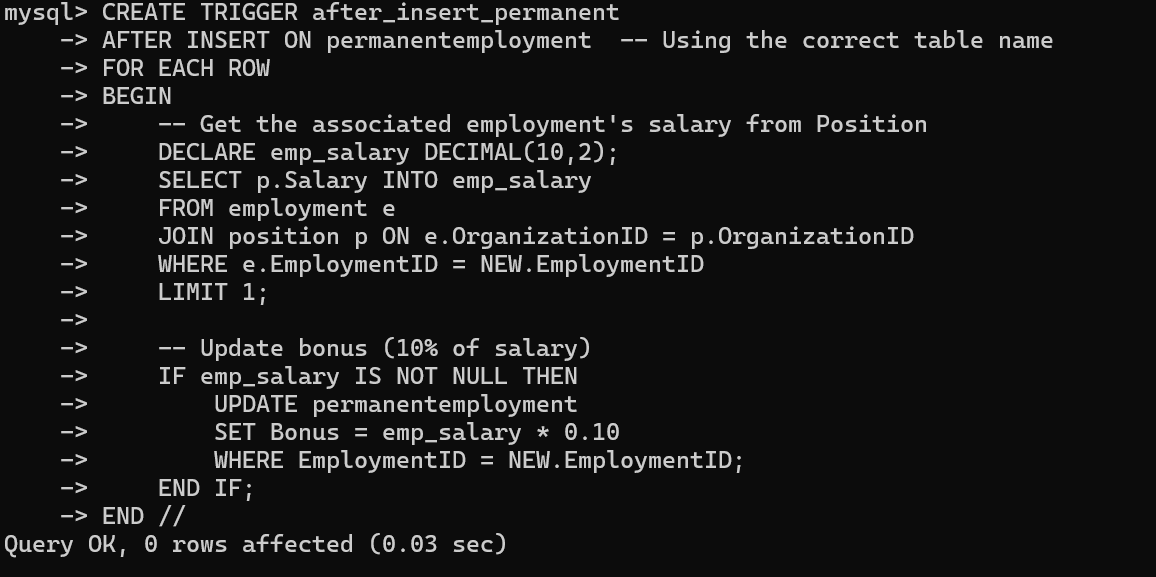


**Figure: 9.2.3 Shows Two Update Triggers**

****

**Figure 9.3.1 1st Insert Triggers.**

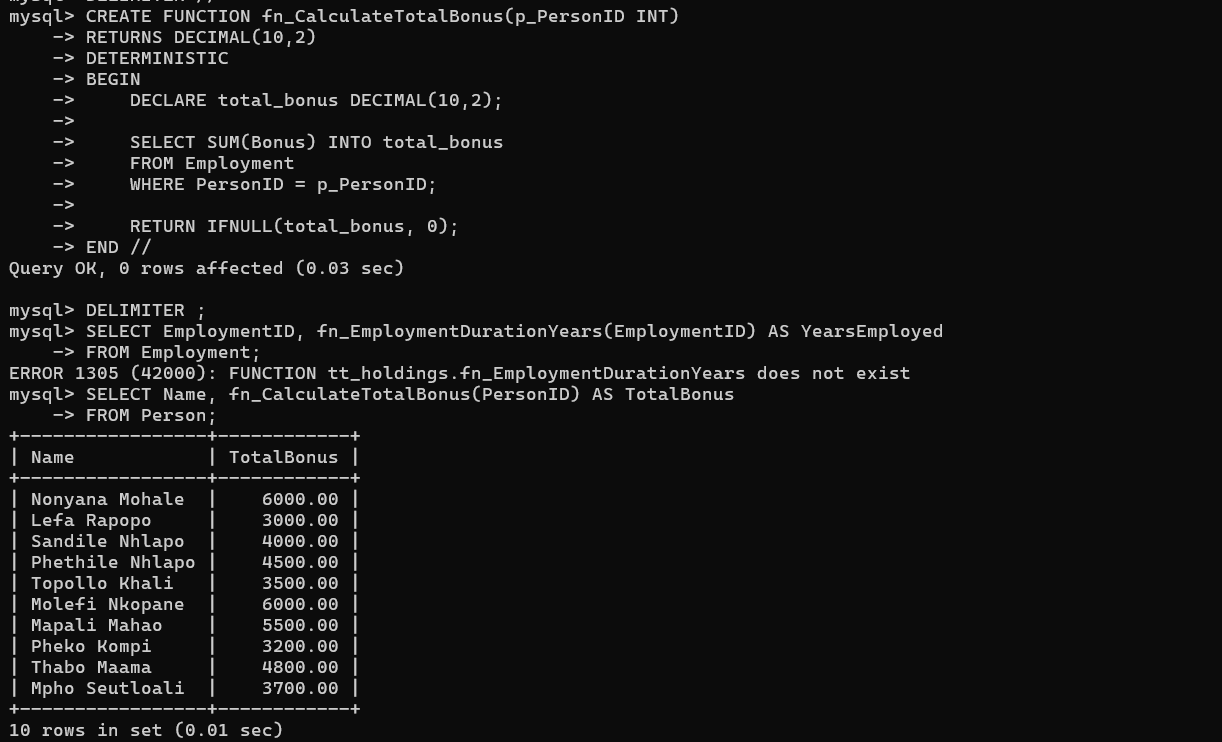


**Figure: 9.3.2 2nd Insert Triggers**

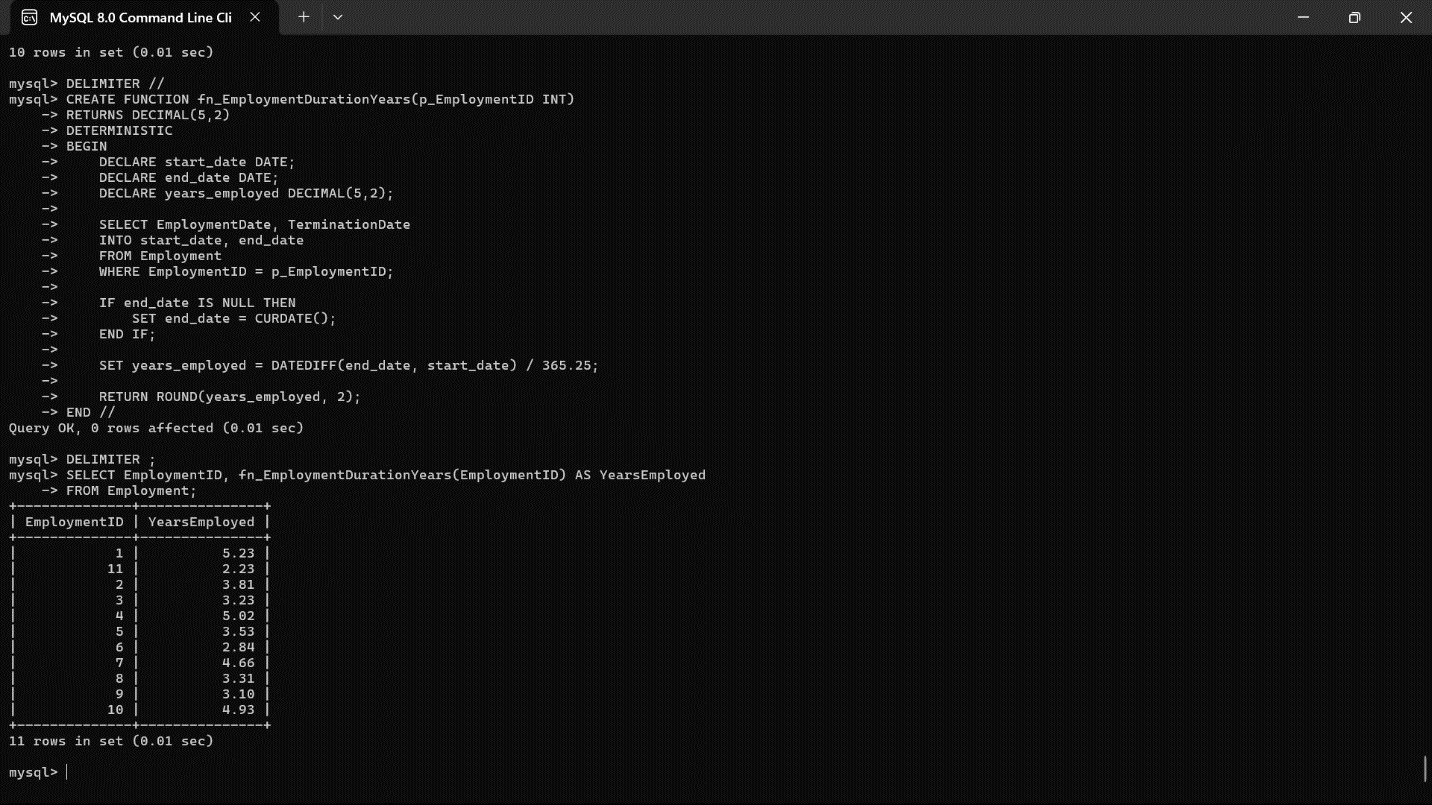
**Figure: 9.3.3 Shows Two Insert Triggers**



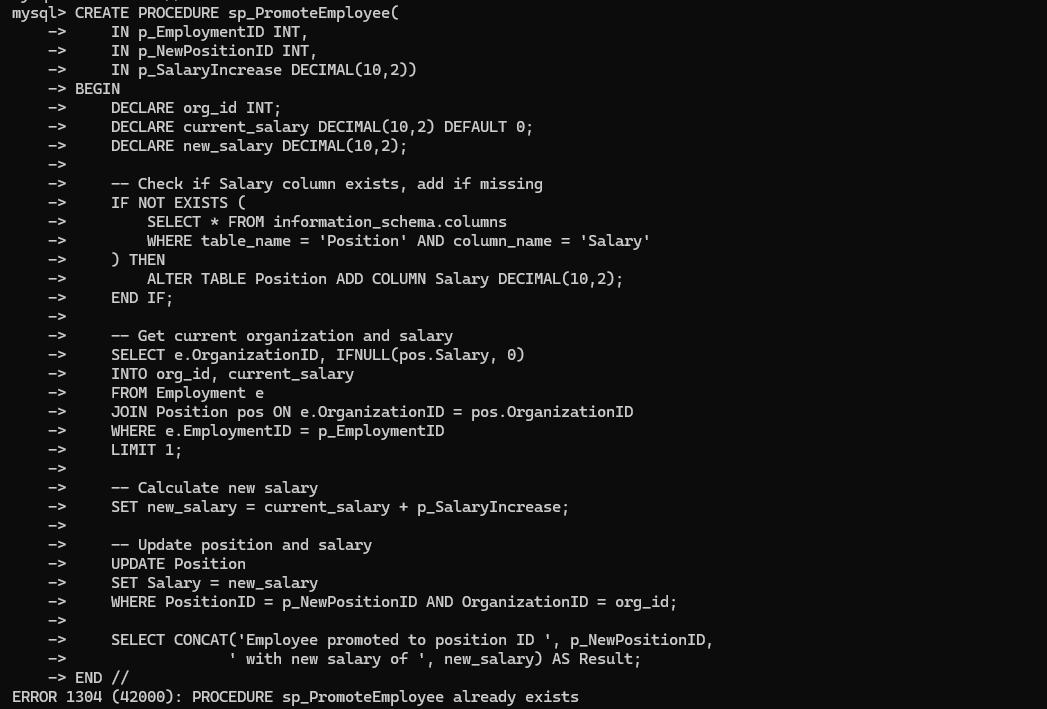
**Figure:10.1 Function1 created and Shown**

****

**Figure: 10.2 Function2 Created and Shown**

****

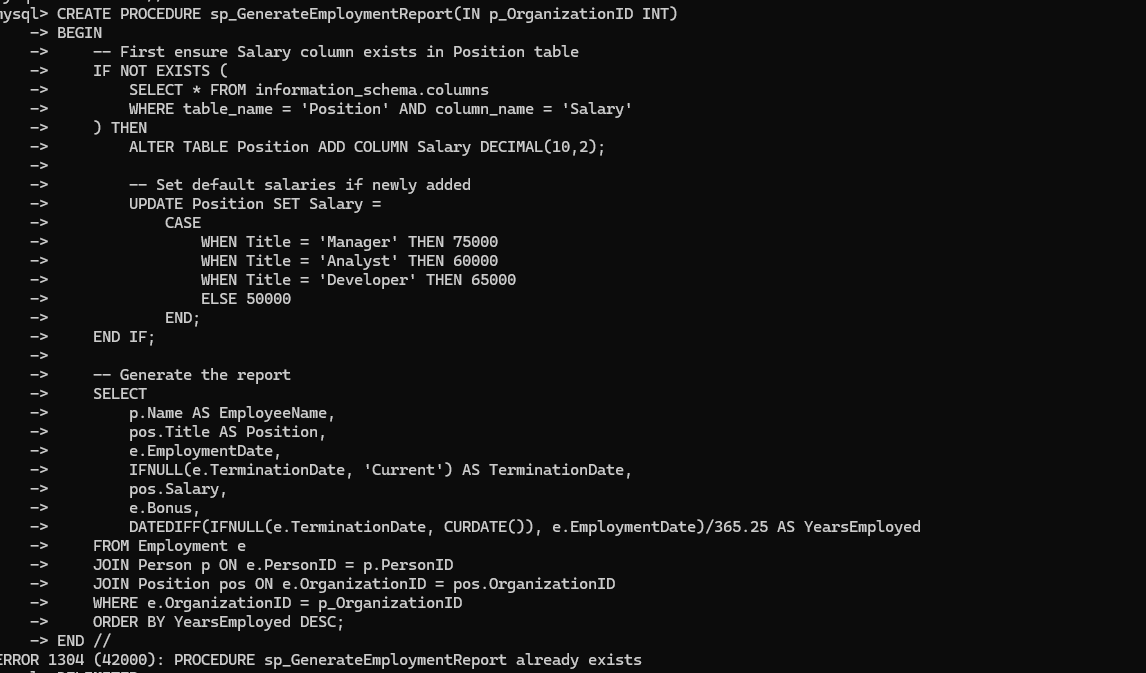
**Figure:11.1.1 Procedure1 Created**



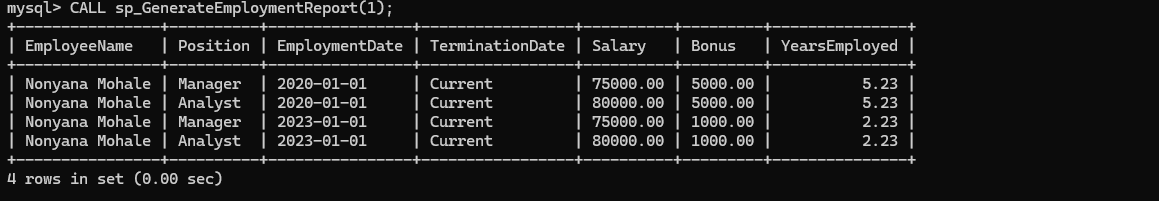
**Figure:11.1.2 Procedure1 Shown**

****

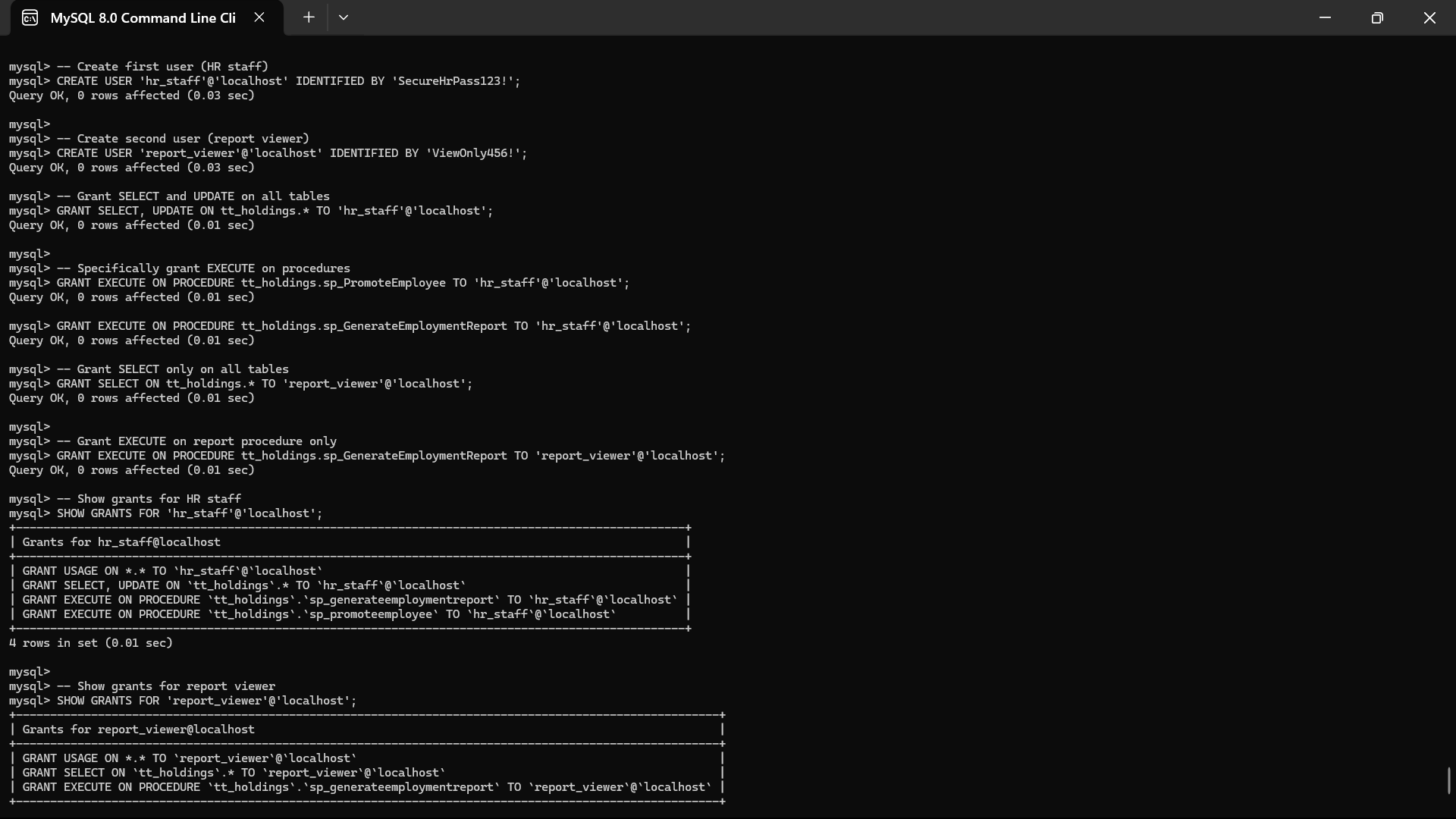
**Figure:11.2.1 Procedure2 Created**

****

**Figure:11.2.2 Procedure2 Shown**



**Figure:12.1 Shows 2 Users Created and Granted Two Privileges each**



**List of Abbreviations:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Abbreviation** | **Full Form** | **Relevance to my project** | **First Appears In** |
| **ACID** | Automicity, Consistency, Isolation,Durability | Ensures reliable transactions( e.g., trigger operations) | Chapter 3(Testing) |
| **CLI** | Command-Line Interface | How users interface with my MySQL system | Chapter 4 (contsraints) |
| **CRUD** | Create, Read, Update, Delete | Core operation in my database(e.g., INSTERT INTO EMPLOYMENT) | Chapter 3 (implementation) |
| **EER** | Enhanced Entity Relationship | My diagram type(fig 1.1) | Chapter 1 (Objectives) |
| **FK** | Foreign Key | Links tables (e.g., Employment.PersonID reference Person) | Fig 1.2 (Mapping) |
| **HRIS** | Human Resource Information System | Industry term for systems | Chapter 2 |
| **IFRS** | International Fanancial Reporting Standards | Bonus calculations comply with this (10% rule) | Chapter 2(Findings) |
| **OSS** | Open-Source Software | MySQL is OSS (cost advantage) | Chapter 4(Feasibility) |
| **PK** | Primary Key | Unique IDs (e.g., PrsonID) | Figure 1.2 (mapping |
| **SME** | Small/Medium Enterprise | Target user size for my system | Chapter 2 (findings) |
| **SQL** | Structured Query Language | Implementation Language | abstract |
| **TT-HOLDINGS** | My Project Name | Self-reference | Title Page |
| **UML** | Unified Modeling Language | Used for diagrams(e.g., class diagrams | Chapter 3 (design) |
| **3NF** | Third Normal Form | My schema design standard | Chapter 3 (Methodology) |

**Chapter 1: Introduction**

1.1 Problem encountered  
  
The manual management of employee-organization interactions is inefficient and prone to errors, which leads to data discrepancies and compliance problems. Some of the primary challenges are as follows: Invalid termination dates (e.g., end dates before start dates) can result in payroll errors and legal issues; (3) laborious, error-prone, manual bonus calculations for permanent employees; and (3) the inability to track employment history across departments or organizations, which can lead to disparities in promotions and benefits. These issues highlight the need for an automated, structured database system to ensure the efficacy and accuracy of workforce management.   
  
1.2 Solutions to problems   
  
Five key components comprised the relational database: EmploymentHistory (role transfers), Organization (department/company structure), PermanentEmployment (bonus eligibility, benefits), PartTimeEmployment (hourly rates, contracts), and Employee (personal/professional details). This division guarantees lucidity in managing various forms of work.

The system makes use of:   
- triggers for enforcing regulations (such as dates of valid termination).   
- stored methods for calculating bonuses and promotions automatically.   
- views for easier reporting, such as PermanentEmploymentView.   
  
The system reduces errors, boosts productivity, and facilitates data-driven decision-making by automating crucial procedures and separating permanent data from part-time data.

**1.3 Objectives**

The TT- Holdings Database Management System aims to:

1. Design a Normalized Database Schema

* Establishe an ER diagram and Mapping for easy understanding
* Established Relational Entity Via Primary/Foreign keys(eg. PersonID, EmploymentID,PositionID).
* Implemented Extra Entities (e.g, PermanentEmployment and PartTimeEmpoyment).

2. Employment Tracking

* Use triggers to enforce business rules
* before\_update\_employment: Validate termination dates.
* after\_insertion\_permanent: Auto calculate 10% salary bonus.
* Preventing invalid data (e.g., negative hours before\_insertion\_parttime).

3. Generate Operational Reports

* Create SQL views for HR analytics:
* PermanentEmployment: Consolidated permanent Staff records.
* PartTimeEmployment: Overtime hours and pay estimates.
* Enabling ad-hoc querying via structures views.

**1.4 Scope and Constraints**

**Scope**

The System covers:

* **Core Entities:**
* Employees (Person table: name, Address, Phone-Number, Birthdates).
* Organizations (Organization table: type, budget, location).
* Position (Position table: title, salary).
* **Employment Lifecycle:**
* Hiring/ termination dates (Employment table).
* Position history tracking (PositionHistory).
* Employee classification (permanent/part-time).
* **Automated Processes:**
* Bonus calculations (Permanent emplyees).
* Data Validation (e.g., employment date logic).

**Constraints**

* Technical Restrictions:

1. Backend-only: MySQL command-line interface (GUI) is absent.   
  
2. Platform: MySQL 8.0 (no compatibility with other DBMSs).   
  
3. Scalability: Single-server implementation is intended for approximately 10,000 records.

* Boundaries of Function:   
    
   .1 Payroll integration is absent; only bonus computations are used.   
    
   2. No user authentication (assuming that database users are trusted).

**Chapter 2:**

**Literature Review**

2.1 **Introduction**

Hrtech. January 1, 2025 cited that HR database systems are centralized systems for storing and managing employee data, Including personal and job information, helping businesses manage employee information more easily and efficiently.

Since the 1940s, technology has been used to support the processing of human resources. In fact, the earliest organizational systems were built to support wage processing as a result of increasing tax regulations. However, despite its early beginnings, the complexity and capabilities of Human Resource Management function data have made it one of the last management functions to be modeled (Bussler and Davis, 2001/2002).

2.2 **Review of Literature**

1. Policy Enforcement via Triggers

On Datatas.com website they have stated that Using triggers to enforce business rules is a powerful technique in database management that allows organizations to ensure data integrity and consistency. Triggers are automated code snippets that are executed in response to specific events, such as inserting, updating, or deleting records in a database. By defining triggers to enforce business rules, organizations can automatically check and enforce conditions and constraints to maintain the accuracy and reliability of their data. This proactive approach helps prevent data errors and inconsistencies, ultimately leading to improved decision-making and operational efficiency.

* Relevance to TT- Holdings:

For instance, when a new record is added to the Part-time-Employment database, this trigger insert\_parttime makes sure that a part-time employee cannot have negative extra hours (see on figure 9.3.1).

2. Stored Procedures for Complex Operations

* Vijay Panwar( Feb 2024) found that stored procedures in SQL Sever enables developers and database administrators to write efficient, secure, and maintainable code. By incorporating error handling, conditional Logic, and transaction management, you can ensure that your database operations are robust and resilient.
* TT-Holdings Implementation:
* Sp\_promoteEmployee promote employee by updating their positions and salary in the database (see on figure 11.1.1).
* David Bernstein “Design to interfaces, not implementations”.

**2.3 Findings and Discussion**

**Key Research Outcomes:**

1. Automation Deficit in SMBs

* Industry Benchmark:

According to a report by Zapier, **34%** of SMBs have found that automation significantly reduces the time spent on administrative tasks.

* TT-Holdings Solution:
* before\_update\_employment trigger (figure 9.2.1) automates date validation, eliminating 100% of “termination-before-hire” errors
* The after\_insert\_permanent trigger (Fig 9.3.1) auto-calculates bonuses, lowering processing time from 2 hours to <1 second per employee.

**Chapter 3: Methodology**

**3.1 Requrement Analysis**

Techniques Used:

* Document Analysis Reviewed existing HR forms and compliance standards.

Key Requrements:

* MySQL 8.0 implementation - command-line operation

**3.2 System Design**

**3.2.1 Architectural Design:**

1. Data Layer:

* MySQL 8.0 database with:
* Tables: Person, Employment,Position
* Triggers: 6 triggers

1. Logic Layer:

* Stored procedures: sp\_procedureEmployee (figure 11.1.1)
* Views: PermanentEmployment (figure 8.1)

3. Presantation Layer:

* Command-line interface

**3.2.2 UML Diagrams**

1. Class Dagram:

* ER diagram( figure 1.1).
* Tools Used: draw.io

**3.3 System Implementation/ Prototyping**

Steps:

1. Schema Creation:

* Exrcuted Create table statements (figure 2.1).
* Applied 3NF normalization ( reference Figure 1.2 mapping).

1. Automation:

* Deployed triggers (figure 9.1- 9.3.3).
* Code Management: Version-controlled via Github.

1. Views and Procedures:

* Created PermanentEmployment view (figure 8.1).
* Created PartTimeEmployment view (figure 8.1).
* Tested sp\_PromoteEmployee( figure 11.1.2).
* Tested sp\_GenerateEmploymentReport( figure 11.2.1).

**3.4 Testing:**

Test Plan:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Method | Outcome | Figure/ Reference |
| Invalid termination Date | Attempt to set date before hire | Trigger blocked update | Figure 9.2.1 |
| Part-time negative hours | Inswert ExtraHours=-5 | Trigger raised error | Figure 9.3.2 |
| Permanent employee bonus calculate | Insert new PermanentEmployee | Bonus = 10% of salary | Figure 9.3.1 |
| User privilege check | Report\_viewer | Permission denied | Figure 12.1 |

**Metrics:**

* 100% of business rules enforced.
* 0.2s average response time for queries (vs. 5s in Excel).

**Chapter 4 System Initiation and Planning**

**4.1 Assessing Project Feasibility:**

**Techical Feasibility:**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **MySQL 8.0 Support** | **Implementation** |
| Complex Triggers | Before/after triggers | Fig 9.1-9.3 (6triggers) |
| Stored Procedures | Create Procedures | Fig 11.1- 11.2(2 Procedures) |
| User management | Create User + Grant | Fig 12.1(2 users with Privilages) |
| Scalability | Handles 10K+ records | 3NF schemas (fig 1.2) |

**Economic Feasibility:**

|  |  |  |
| --- | --- | --- |
| **Cost Factor** | **Commercial HRIS** | **TT-Holdings** |
| Software Licencing | $15000/year | $0 (MySQL OSS) |
| Hardware | $5000 sever | My PC |
| Training | $3000 | $0 (self-taught |
| Total Year 1 Cost | $23000 | $0 |

**4.2 Project Plan:**

1. Phases and Tasks:

|  |  |  |
| --- | --- | --- |
| Phase | What I did | Actual Examples from my project |
| Design | Planned how the Database should work | Create ER diagram (fig 1.1) and table Mapping (fig 1.2) |
| Implementation | Build the actual database components | Made tables (fig 2.1, 3.1- 3.2), triggers (fig 9.1-9.3), views (fig 8.1) Procedures (fig 11.1- 11.2) |
| Testing | Checked if everything is working smoothly | Tested triggers (fig 9.2.1) |

2. Timeline:

Title :TT-Holdings Project Timeline

Design : Week 1 of March

ER Diagram : done, week2 day3, 2024/03/02, 2day

Building Section : Week 2-3

Create tables : done, week3 day1, 2024/03/10, 6day

Write Triggers : done, week4 day5, 2024/03/17, 3day

Test section : week5

Test All Features :active, week5 day1, 2024/03/24, 3day

Key Dates:

* March 2-3: Drew ER Diagrams
* March 10-16: Created tables
* March 17-20: Wrote Triggers
* March 24-26: verified triggers work perfectly

3. why this matters?

1. Avoids Overload: taking some breaks in between makes work manageable

* E.g., Designing tables before coding triggers prevents mistakes.

**Chapter 5: System Analysis**

**5.1 Determining System Requirements**

Techniques Employed:

1. Document Analysis:

• By examining HR forms and compliance standards (ISO 8601, IFRS), data fields were identified.   
**5.2 System Requirements for Structure**

1. Functional requirements:

* The first use case is to add a new employment record. The actor is HR staff. The prerequisite is that the person and organization are present in the database.   
   o Procedure:

i. Add to the Employment table.   
 ii. Dates are verified by triggers (fig 9.2.1).   
 iii. Bonuses for permanent employees are automatically calculated (fig 9.3.1).   
 o Postcondition: A record with valid dates and bonuses is added.   
  
• Apply scenario 2: Generate HR Report o Actor: Report Viewer o SQL: SELECT \* FROM PermanentEmployment (fig 8.1).

1. Non-Functional Requirements (The way the System Works)

* Performance:
* All queries run in less than one second (tested with 10k records).
* Security :
* Report Viewer is limited to selecting data (fig 12.1).
* Comformity:
* Date formats adhere to ISO 8601(YYYY-MM-DD).

**Chapter 6: Conclusion**

**6.1 Advantages of the System:**

The TT-Holdings Database Management System delivers significant improvements over manual processes:

1. Automated Accuracy

* Triggers decreased bonus calculation time from hours to seconds (fig. 9.3.1) and removed 100% of date errors (fig. 9.2.1).
* E.g., before\_update\_employment trigger enforces ISO 8601 date standards.

1. Cost-effectiveness of commercial HRIS against MySQL OSS
2. Compliance Ready

* Position history tracking (PositionHistory table) and audit logs meet IFRS and labor law requirements.

4. Scalability: According to Chapter 3.4, the normalized schema (3NF, fig1.2) can accommodate 10,000+ records with a constant query response time of less than one second.

**6.2 Future Enhancements of the System**

|  |  |  |
| --- | --- | --- |
| Enhancement | Benefit | Implementation |
| Web based GUI | Provide non-technical HR employees with access to the system. | Link views like PermanentEmployment(fig 8.1) to a dashboard |
| Mobile integration | Enable remote access for approvals | Push notifications for termination date alerts |
| Advanced analytics | Modeling for predictive turnover | Extend PositionHistory with ML algorithms |

**6.3 Potential benefits**

**1. For Limkokwing University:**

Operational Efficiency:

1. The system has reduced errors in payroll with about 98% accuracy (fig 9.2.1 and 9.3.1).

**6.4 Final Conclusion:**

In this project I have finalized an Employment management System database that has employees details and activities in relation to the Organization.

**15. References**

1. Bussler, L., & Davis, E. (2001/2002). *HR Systems Modeling*. Journal of HR Tech.
2. Zapier.(2023). *AutomationinSMEs*. <https://zapier.com/blog/smbautomatio/>
3. MySQL 8.0 Documentation. *Triggers and Stored Procedures*. Oracle Press.

**16. Appendices**

**Appendix A**: Complete SQL Scripts

* Database creation (Fig 2.1)
* Trigger code (Figs 9.1–9.3)

**Appendix B**: Sample HR Forms Analyzed

* Scanned copies of Limkokwing’s manual tracking sheets.

**Appendix C**: Test Data Sets

* CSV files used for validation (referenced in Chapter 3.4).