Employment Management System (EMS)

Contents

[List of figures 2](#_Toc193693149)

[List of tables 4](#_Toc193693150)

[list of Abbreviations 5](#_Toc193693151)

[Abstract 5](#_Toc193693152)

[Chapter 1: Introduction 5](#_Toc193693153)

[1.2 Problem Solving 5](#_Toc193693154)

[1.3 Objective 6](#_Toc193693155)

[1.4 Scope & Constraint 7](#_Toc193693156)

[Chapter 2: Review of Literature 7](#_Toc193693157)

[2.1 Introduction 7](#_Toc193693158)

[2.2 Reviewing the Literature 7](#_Toc193693159)

[2.3 Findings & Discussion 8](#_Toc193693160)

[Chapter 3 8](#_Toc193693161)

[3.1 Requirement Analysis 8](#_Toc193693162)

[3.2 System Design 9](#_Toc193693163)

[3.2.1 Architectural Design 9](#_Toc193693164)

[3.3 System Implementation / Prototyping 9](#_Toc193693165)

[3.4 Testing 11](#_Toc193693166)

[Chapter 4: System Initiation and Planning 11](#_Toc193693167)

[4.1 Assessing Project Feasibility 11](#_Toc193693168)

[4.2 Project Plan 11](#_Toc193693169)

[Chapter 5: System Analysis 11](#_Toc193693170)

[5.1 Determining System Requirements 12](#_Toc193693171)

[5.2 Structuring System Requirements 14](#_Toc193693172)

[Chapter 6: Conclusion 15](#_Toc193693173)

[6.1 Advantages of the System: 15](#_Toc193693174)

[6.2 Future Enhancements of the System 15](#_Toc193693175)

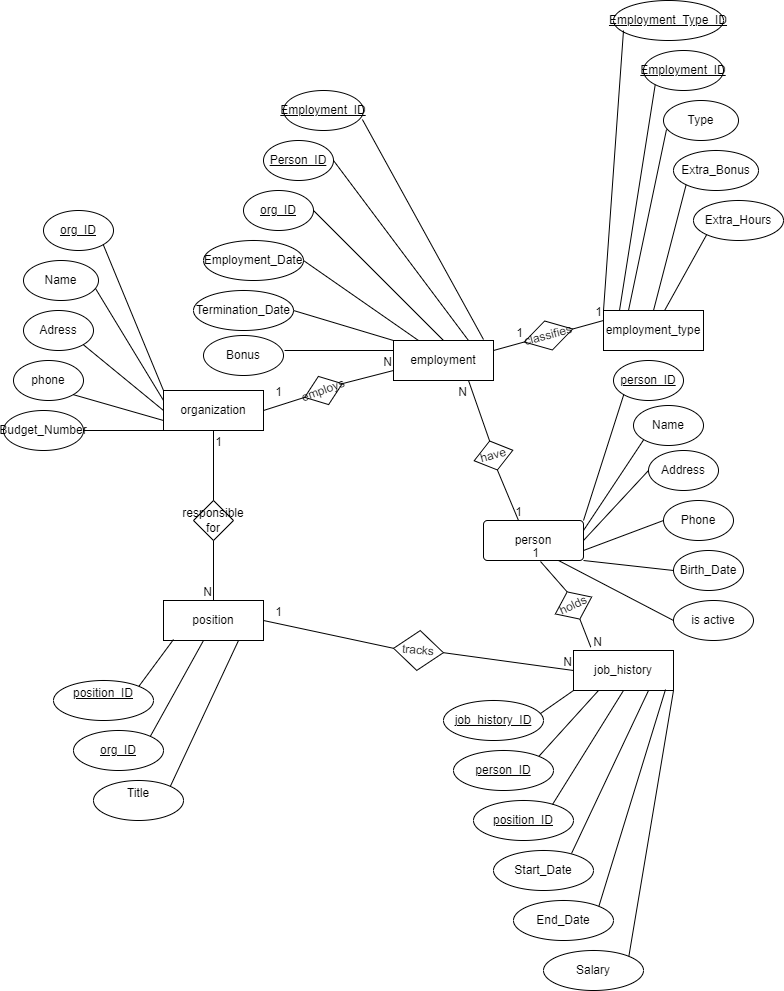
[6.3 Potential Benefit 15](#_Toc193693176)

[6.4 Conclusion 16](#_Toc193693177)

[REFERENCES 16](#_Toc193693178)

[Appendices 17](#_Toc193693179)

# List of figures

****

# List of tables

employment

employment\_type

job\_history

organization

person

position

**NB:** I have another tables the ones I ran as I was creating views, triggers, functions and procedures as to test and see if my project work properly. Namely:

active\_employees

active\_jobs

employment\_less\_than\_10

employment\_type\_log

expert\_positions

high\_bonus\_employees

high\_budget\_organizations

high\_org\_positions

high\_salary\_jobs

hlotse\_maputsoe\_residents

older\_than\_1995

organization\_update\_log

permanent\_employees

person\_log

person\_update\_log

# list of Abbreviations

ER-Entity Diagram   
SQL is a structured query language.  
The PK-primary key  
Fk-foreign key   
Employment Management System (EMS)

## Abstract

This report presents the development of an employment and organizational data management system to facilitate efficient tracking of person, employment history, organizational roles, and compensation. The project incorporates a structured approach using ER diagram, relational database modeling, and SQL functionalities such as views, triggers, functions, and stored procedures. Security and user access control mechanisms are also implemented to ensure data integrity and privacy.

Chapter 1: Introduction 1.1 Problem Statement

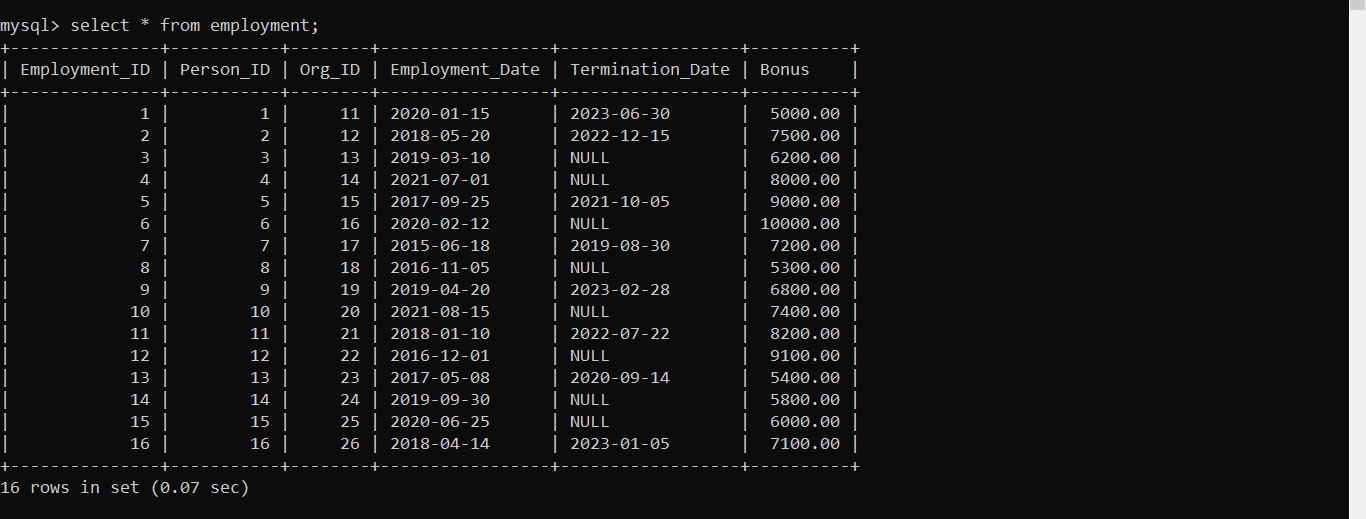
Effective management of employment data is difficult for organizations, especially when people may work for several companies or occupy several positions at the same time. According to Elmasri & Navathe (2016), inefficiencies, tracking errors, and compliance problems result from current systems' frequent inability to capture the intricacy of these interactions.

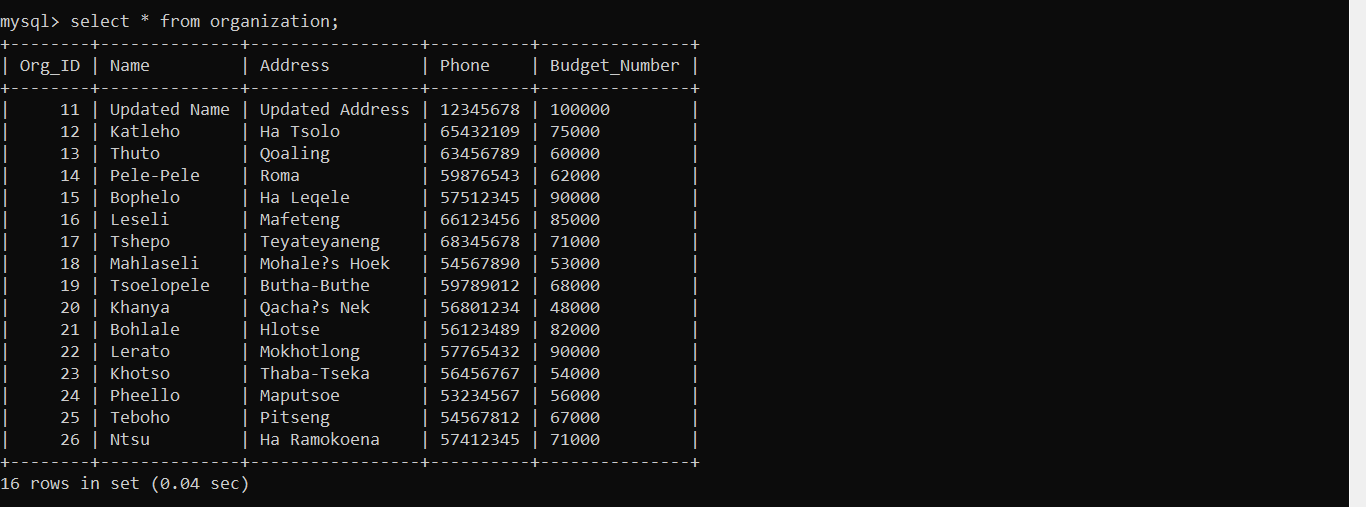
## 1.2 Problem Solving

The EMS aims to:

* Establish a structured database for managing employment and organizational records.
* Provide a relational framework for tracking job roles, salaries, and work history.
* Implement security measures through user role-based access controls.
* Automate data operations using advanced SQL techniques

**Below are pictures to display employment and organizational records**



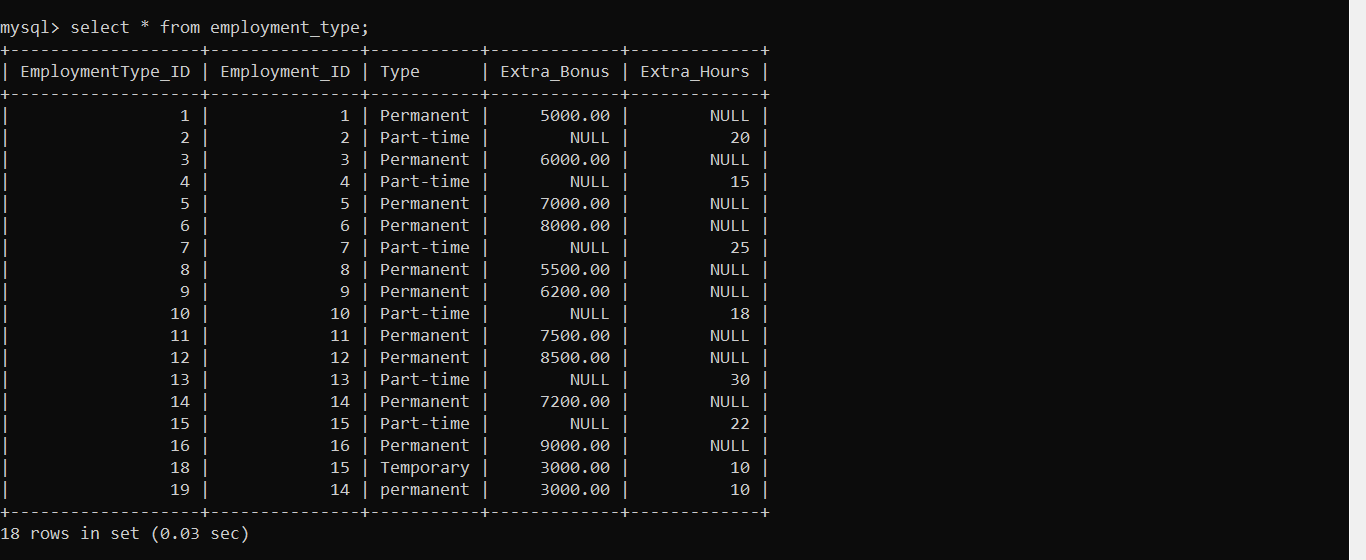


## 1.3 Objective

The main goals of the EMS are:

* To create a comprehensive database (ttemployment) that captures all necessary individual and organizational information, such as IDs, names, addresses, and contact details.
* To differentiate between employment types (permanent and part-time) and manage associated bonuses and salary details effectively.
* To track detailed job histories and position assignments for individuals across multiple organizations.

**Here is a well displayed part that shows an employment\_type**

****

## 1.4 Scope & Constraint

The EMS encompasses functionalities for person and organization management, employment tracking, position management, and job history documentation. Constraints include potential data integration challenges, varying organizational policies, and the need for user training.

# Chapter 2: Review of Literature

## 2.1 Introduction

According to findings by Elmasri & Navathe (2016) and Kavanagh & Thite (2009), database management systems are essential for managing structured data. Additionally, they maintain that by guaranteeing organized record-keeping, employment databases are essential to HRM and corporate analytics.

## 2.2 Reviewing the Literature

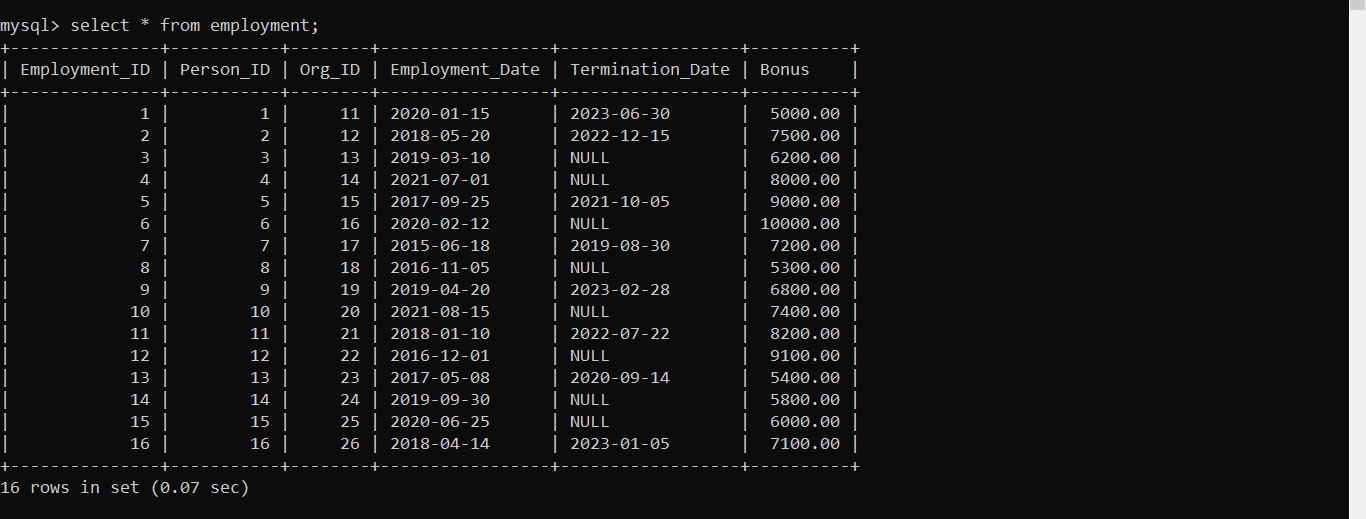
* Employment Management Systems: Studies show that effective employment management systems can increase operational efficiency and accuracy in HR processes. For example, Johnson (2023) emphasized the role of integrated systems in improving data retrieval and management.
* Multi-Organizational Employment: Research by Smith (2022) demonstrates that organizations that manage external and internal employment effectively see higher employee satisfaction and retention rates.

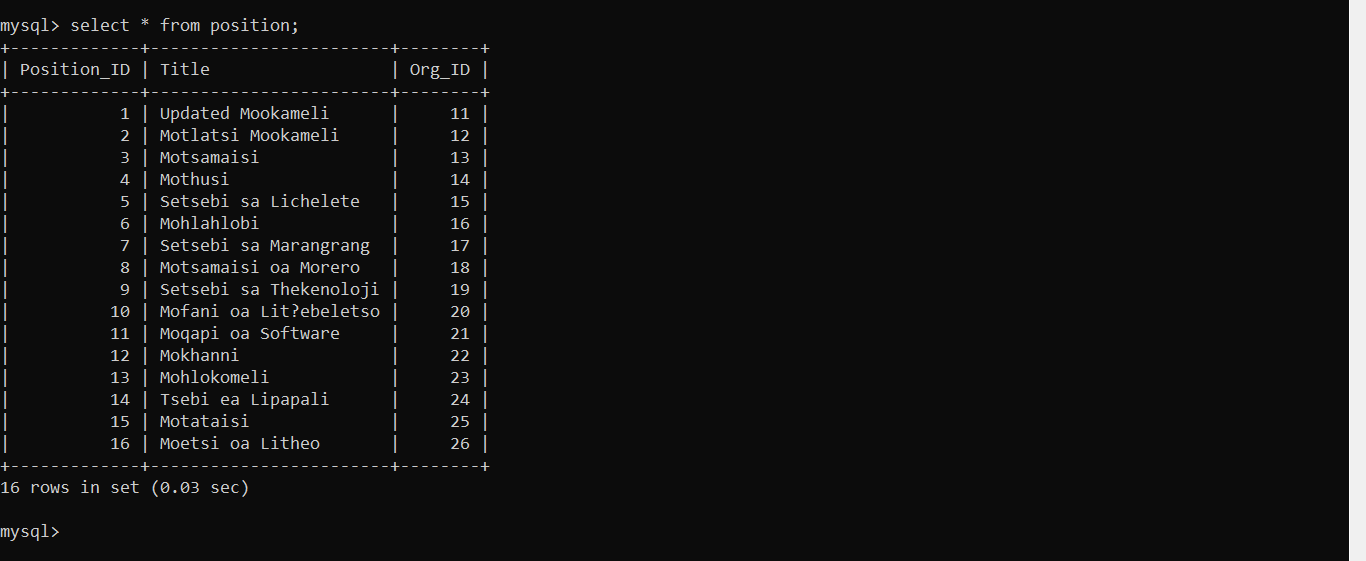
Review of Existing Research

C. J. Date (2000). "A Brief Overview of Database Systems."

## 2.3 Findings & Discussion

Relational databases use normalization to organize data and avoid repetition. According to Priya P. (2023), primary and foreign keys help connect different tables, preventing duplicate data. This makes relational databases more accurate compared to other database systems. The following two-table example shows how these keys link tables together.





Chapter 3**: Methodology**

## 3.1 Requirement Analysis

Key requirements identified include:

* Capturing basic information (ID, name, address, phone, birth date for persons; ID, name, address, phone, and budget for organizations).
* Tracking multi-organizational employment for individuals.
* Documenting salary and bonuses for different employment types.

## 3.2 System Design

## 3.2.1 Architectural Design

Kleppmann (2017) explains that data plays a key role in modern system design challenges. Issues like scalability, consistency, stability, efficiency, and maintainability need to be addressed. There are also many technologies available to manage data, including message brokers, stream and batch processors, relational databases, and NoSQL datastores.

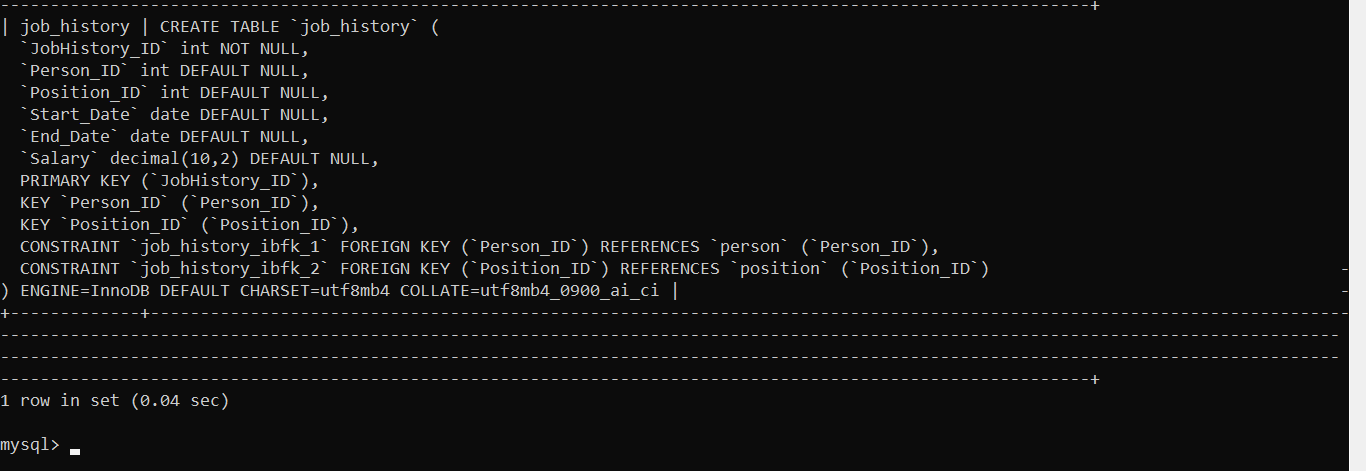
Architectural design comprises three layers:

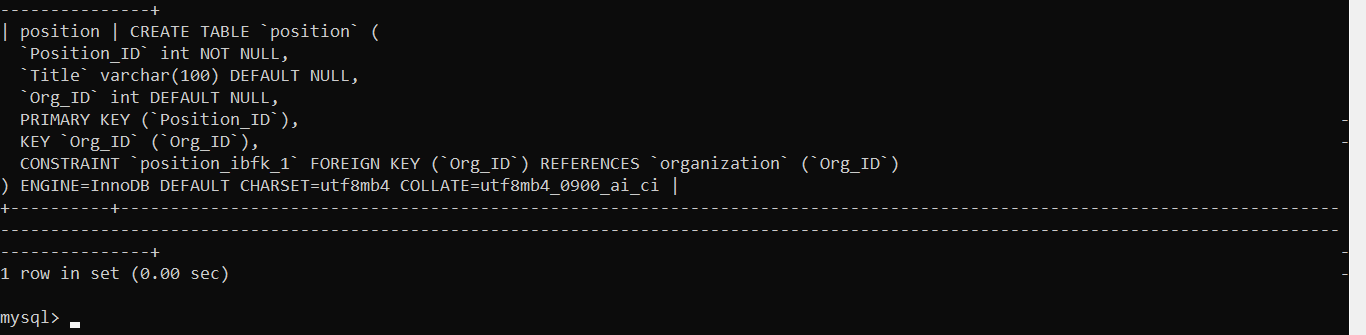
* Presentation Layer: The user interface where HR staff can manage employee and organization records.
* Application Layer: Contains the business logic that processes user requests and interacts with the ttemployment database.
* Data Layer: A relational database (MySQL) specifically designed for the ttemployment database that stores employee and organizational data, structured according to the identified entities.

## 3.3 System Implementation / Prototyping

Development phases include:

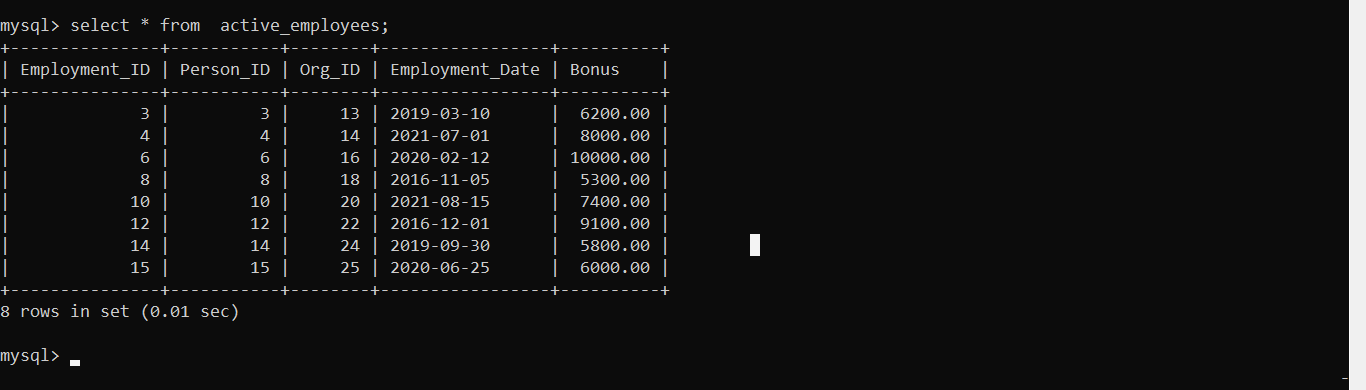
* Creating and configuring database tables.





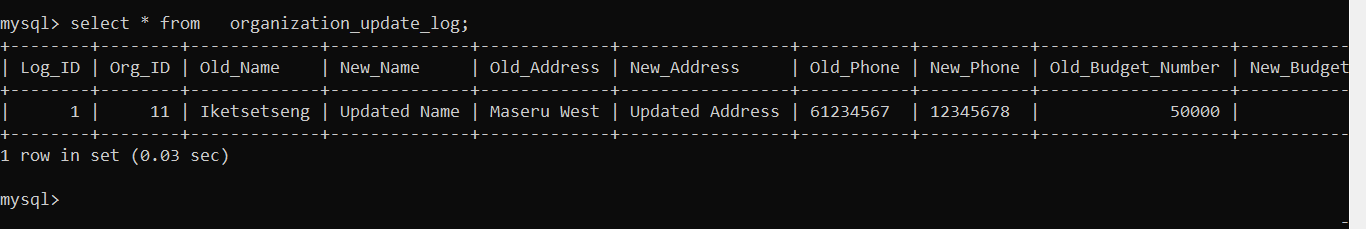
Above are pictures that show how I created my two tables for job history and position

* Implementing SQL-based functionalities such as views, triggers, and procedures.



In this picture is a table view named active\_employees from the employment table which has now hidden the column for Termination\_Date as to prove functionality

The following is a trigger and its table is named “organization\_update\_log”



The last part is for procedures. Here is a picture below for a procedure that shows the functionality

mysql> SELECT \* FROM person WHERE Person\_ID = 3;

+-----------+---------+------------+----------+------------+----------+

| Person\_ID | Name | Address | Phone | Birth\_Date | IsActive |

+-----------+---------+------------+----------+------------+----------+

| 3 | Thabang | Hleoheng18 | 58234567 | 1998-03-10 | 0 |

+-----------+---------+------------+----------+------------+----------+

* Managing user access and security

**NB**: please be aware that some functionalities are left out, so they will be displayed in the appendices

## 3.4 Testing

Testing is critical in software development and ensures the system meets its requirements. Here is where I test function to get personInfo:

mysql> SELECT GetPersonInfo(1);

+--------------------------------------------------------------------------------+

| GetPersonInfo(1) |

+--------------------------------------------------------------------------------+

| Name: Mothibeli, Address: New Address, Phone: 57700000, Birth Date: 1995-01-15 |

+--------------------------------------------------------------------------------+

# Chapter 4: System Initiation and Planning

## 4.1 Assessing Project Feasibility

A feasibility analys determines if a system can expand as needed and is feasible to construct.

## 4.2 Project Plan

The project plan outlines:

* Timeline: A 6-month timeline divided into phases: requirements gathering, development, testing, and deployment.
* Resource Allocation: Listing team members, including developers, data analysts, and project managers.

# Chapter 5: System Analysis

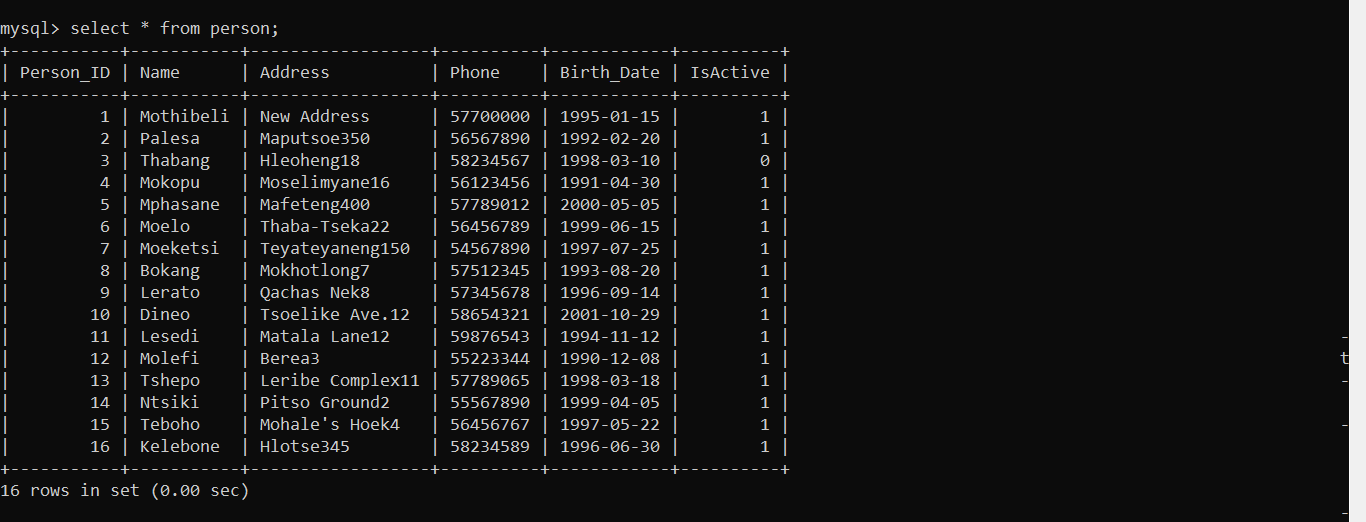
## 5.1 Determining System Requirements

Stakeholder feedback led to identifying the following requirements:

Functional Requirements:

Person Management:

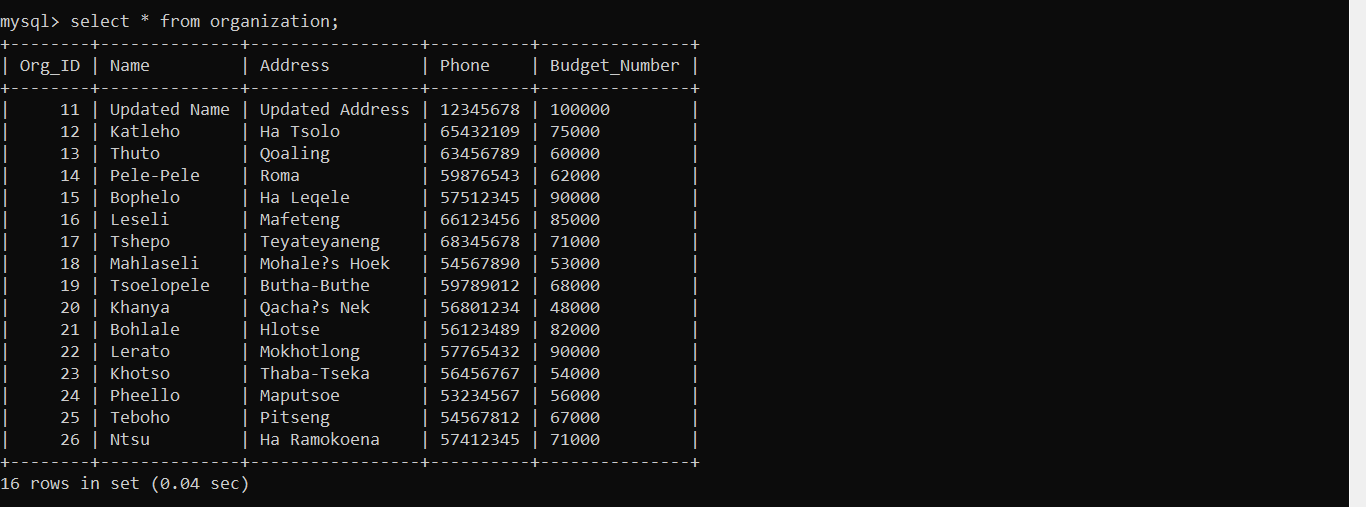
* Store individual data (ID, name, address, phone, birth date).
* Track employment status (active/inactive).
* Maintain employment history and position data.



Here is table for person showing individual data

Organization Management:

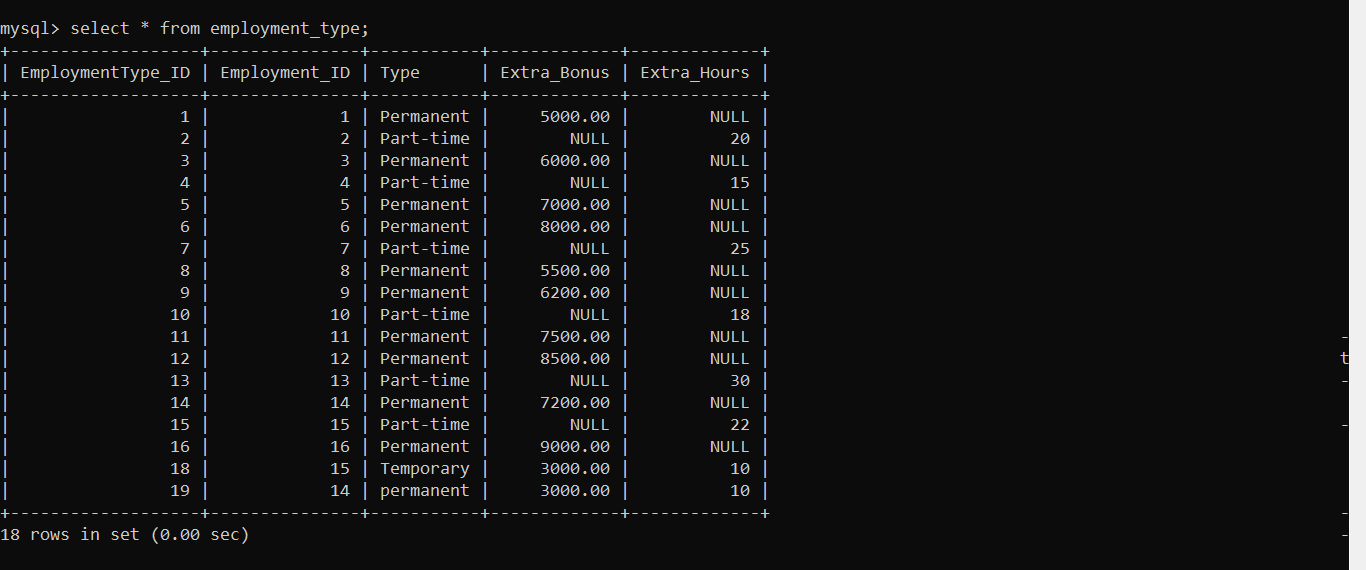
* Store organization data (ID, name, address, phone, budget).
* Track hiring and employment linkages between organizations and persons.



Here organization data is displayed

Employment Tracking:

* Record employment information, including employment date, termination date, and bonuses.
* Differentiate between permanent and part-time employees.



Here is a table that displays data for employment\_type

Position Management:

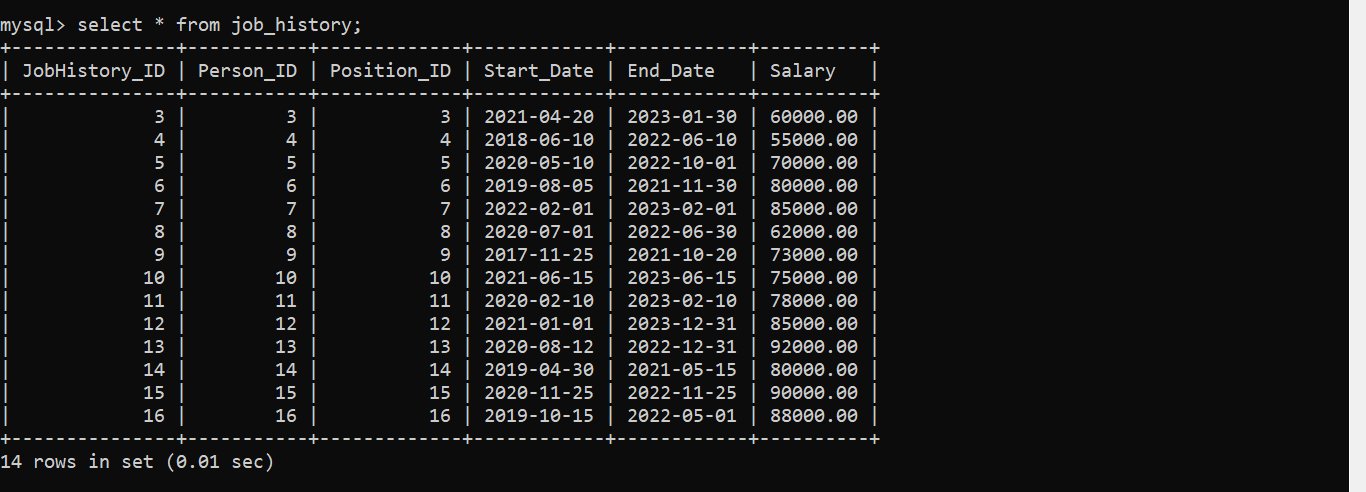
* Store details about position ID, title, organization ID



This is a position table that displays position

Job History Management:

* Track an employee's previous jobs, including job title, employer name, and employment dates.



Non-functional requirements include: • Performance: The system must be able to accommodate 500 users at once without experiencing any performance issues.  
• Security: Role-based access and encryption should be used for all data.   
• Usability: Users should be able to navigate the interface with little assistance thanks to its intuitive design.

## 5.2 Structuring System Requirements

The requirements are grouped into different categories for easier management:

Person Management:

* Create, update, and close user accounts.
* Keep track of job types and their benefits.

Organization Management:

* Create and manage information about the organization.
* Connect people to their roles in the organization.

Employment Management:

* Record job history, including job titles, salaries, and bonuses.
* Make sure job types are correctly identified (like full-time or part-time).

Position Management:

* Define the tasks and skills needed for each job in the organization.
* Connect performance reviews to specific jobs.

Reporting & Analytics:

Create reports about employees, job types, and work history.

# Chapter 6: Conclusion

## 6.1 Advantages of the System:

The Employment Management System has many benefits:

* Efficiency: Automating routine tasks lets HR staff spend more time on important projects.
* Accuracy: Reducing manual data entry lowers the chances of mistakes in employee records, work histories, and payroll calculations.
* Accessibility: A single database makes it easy for authorized users to find employee information, like job history and employment type.

## 6.2 Future Enhancements of the System

To keep the Employment Management System useful and up-to-date, several improvements can be planned:

* Mobile Access: Creating a mobile app so that managers and employees can check their job history and employment details anytime, anywhere.
* Integration with Other Systems: Connecting this system with other HR tools (like payroll and performance management systems) to make data sharing easier.
* AI Analytics: Using AI technology to analyze data and provide insights on employee engagement, predict turnover, and track trends in job history and employment types.

## 6.3 Potential Benefit

By implementing the Employment Management System, organizations can expect numerous benefits:

* Streamlined Operations: A transparent system leads to smoother workflows and less redundancy in employee management processes, especially in tracking job history and employment types.
* Improved Employee Satisfaction: Quick access to personal, employment, and job history information empowers employees and enhances their overall job satisfaction.

## 6.4 Conclusion

Organizations that track employment information have basic needs that are met by the Employment Management System. It improves efficiency and helps companies and employees throughout their careers by consolidating aspects like employment types and job history onto a single platform. Esra NEMLİ ÇALIŞKAN (2010) asserts that an organization's success is directly related to how well it manages its human resources. This bolsters the notion that an organization's ability to compete is contingent upon its allocation of resources towards particular assets. Although research indicates that effective HR practices improve business performance, it's crucial to understand that a variety of other elements also affect an organization's success and that there may be a complex link between HR practices and other resources.

### REFERENCES

Navathe, S. B., & Elmasri, R. (2016). *Database System Fundamentals* (7th ed.). Pearson.

Thite, M., & Kavanagh, M. J. (2009). *Information Systems for Human Resources: Foundations, Uses, and Prospects*. SAGE Books.

Johnson, A. (2023). *By Integrating HR Technologies, Employment Management Systems Can Operate More Efficiently*. HR Tech Press.

Smith, B. (2022). *Managing Employment Across Many Organizations: Techniques for Retaining and Enhancing Employee Happiness*. Workforce Solutions Publishing.

Codd, E. F. (1970). *Data in a Relational Model for Big Shared Data Banks*.

Navathe, S., & Elmasri, R. (2016). *Database Systems Fundamentals*.

Date, C. J. (2000). *A Brief Overview of Database Systems*.

Priya, P. (2023). *Relational Database Systems: Principles and Methods of Normalization*. Publications on Database Research.

Kleppmann, M. (2017). *The Key Concepts Underlying Dependable, Scalable, and Maintainable Systems Are Used While Designing Data-Intensive Applications*. O'Reilly Media.

Thite, M., & Kavanagh, M. J. (2009). *Information Systems for Human Resources: Foundations, Uses, and Prospects*. SAGE Books.

Hendrickson, A. R. (2003). *Information Systems for Human Resources: The Foundation of Modern Human Resources Technology*. *Labor Research Journal*, 24(3), 381–394.

Strohmeier, S. (2020). *Smart HRM: A Delphi Research on the Use and Implications of the Internet of Things in Human Resource Management*. *Journal of Human Resource Management International*, 31(12), 1530–1555.

Boudreau, J. W., & Marler, J. H. (2017). *An Analysis of HR Analytics Based on Evidence*. *International Journal of Human Resource Management*, 28(1), 3–26.

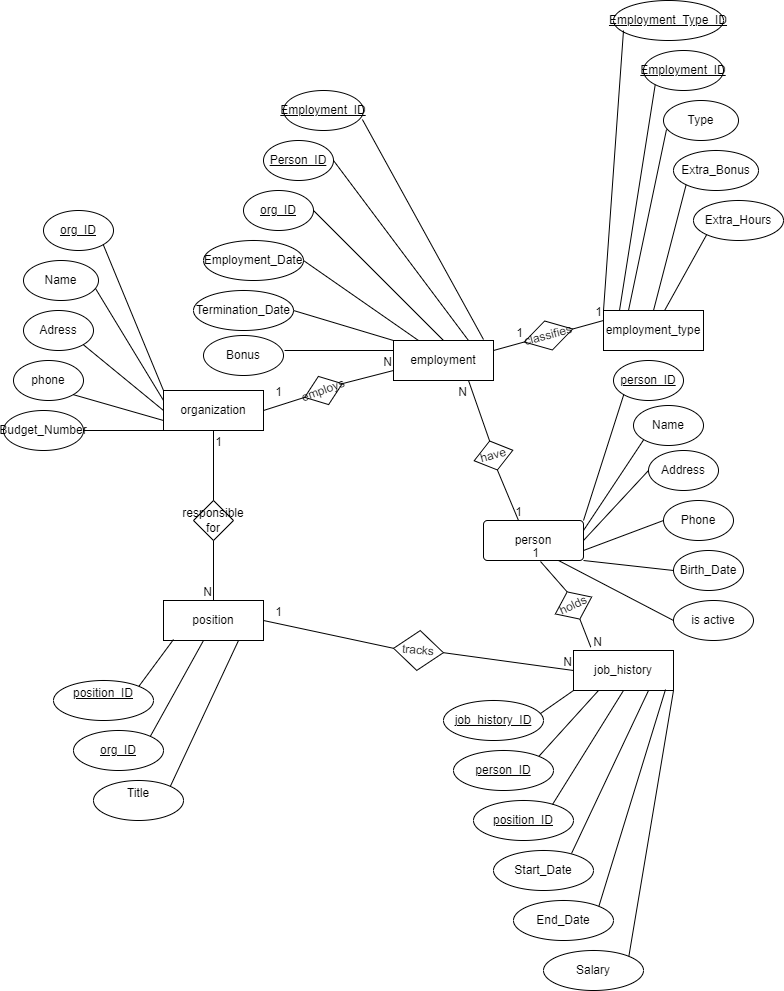
Fisher, S. L., & Marler, J. H. (2013). *A Review of Strategic Human Resource Management and e-HRM That Is Grounded in Evidence*. *Human Resource Management Review*, 23(1), 18–36.

Çalışkan, E. N. (2010). *The Effect That Strategic HRM Has on the Performance of Organizations*. *Journal of Naval Science and Engineering*, 6(2), 100–116.

### Appendices

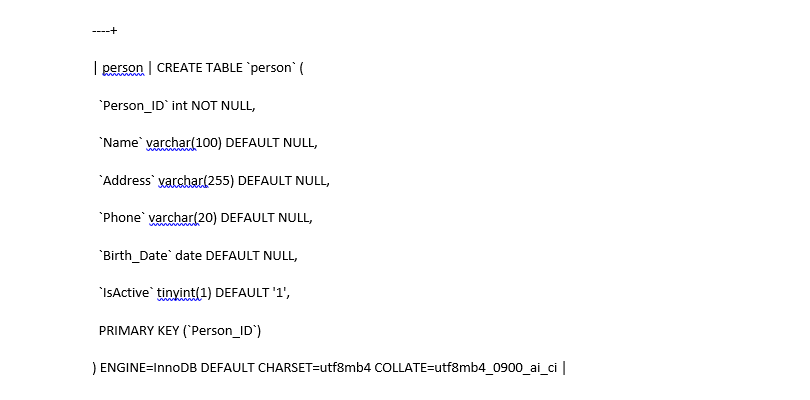
**Appendix A**

ER Diagram

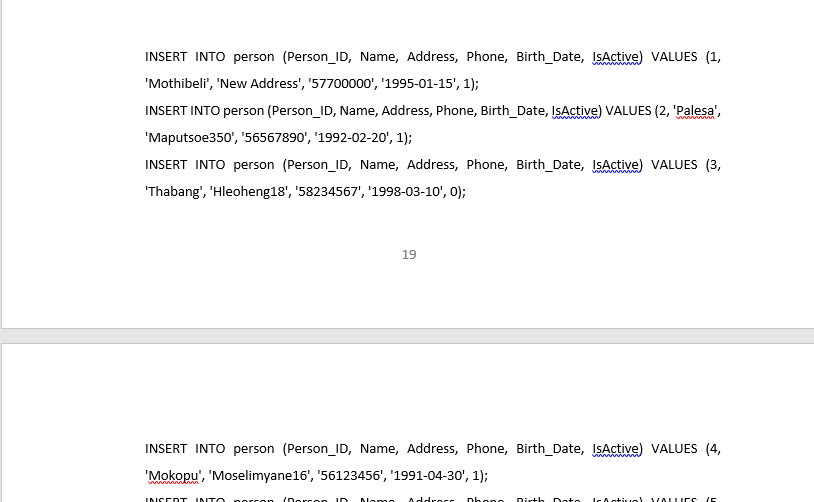


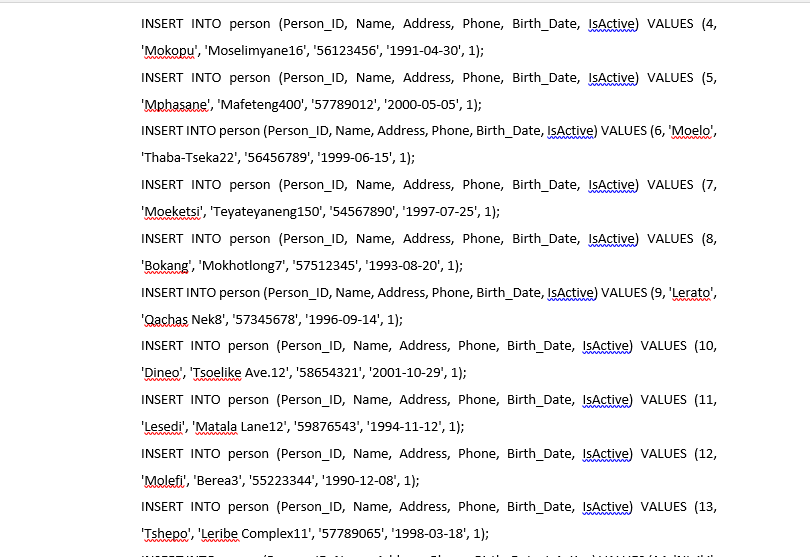
**Appendix B: SQL Commands for Table Creation, Data Insertion, Views, Triggers, Functions, Procedures, Users, and Privileges**

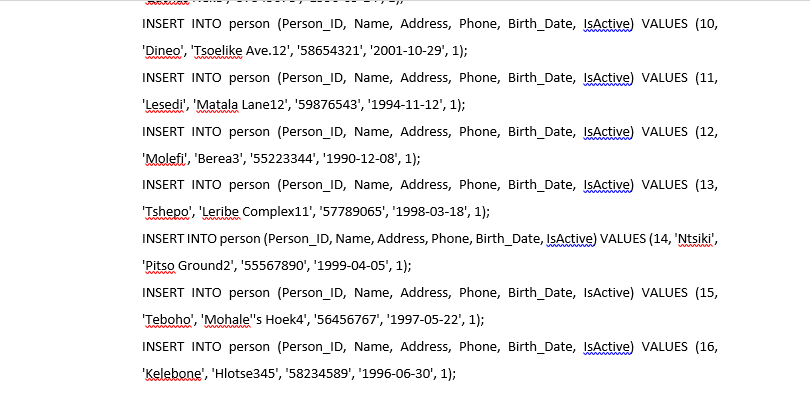
**Table Creation for person**

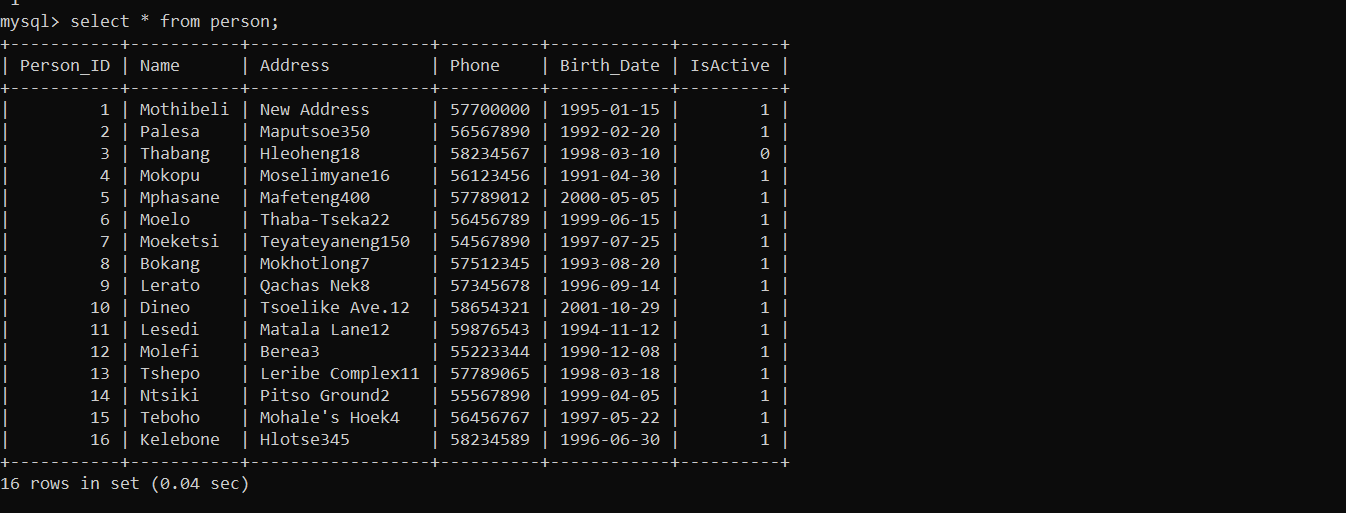


**Data insertion**

****







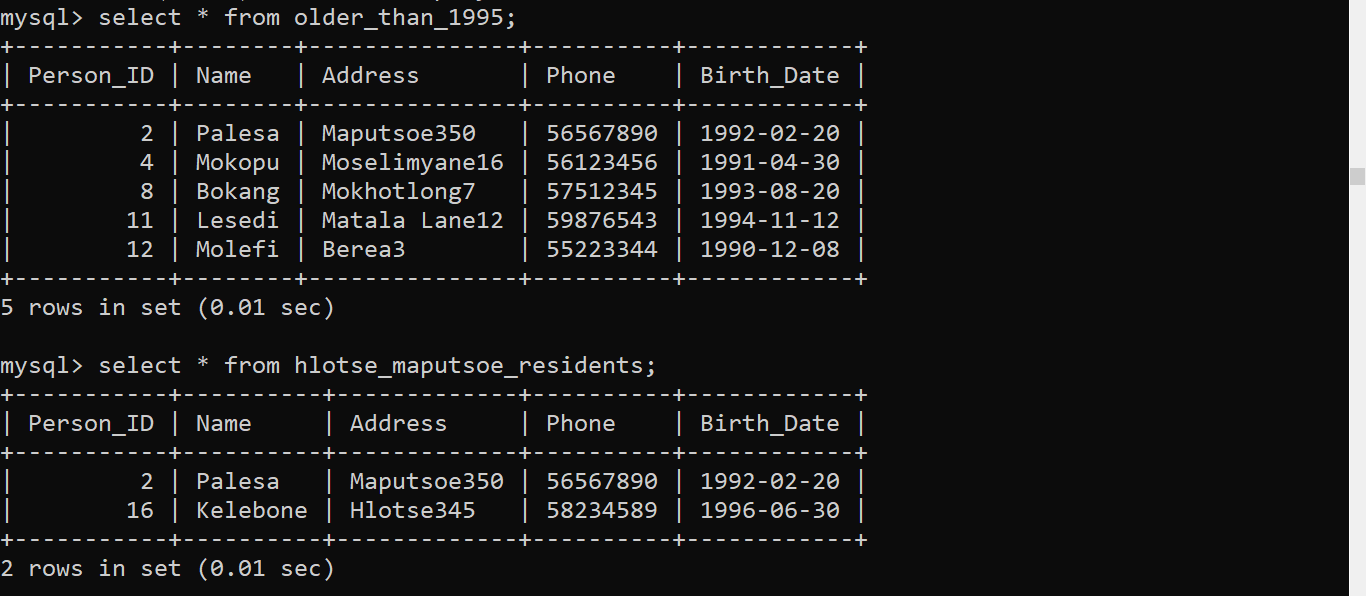
According to Simplilearn (2024), a SQL database view is comparable to virtual tables that hold a single query and its outcome. A virtual table with rows and columns that resemble the real ones is typically the end result. A query can be passed to store all or a portion of a table's rows and columns.

In SQL, a view's primary benefit is that it caches the query, saving us from having to write it repeatedly from scratch. However, the question then becomes, why not adopt a stored process in its place? Even stored procedures have the ability to store a query and run when called. The benefit of views, however, is that they are simpler and easier to understand than procedures.   
Views can only contain one statement, but procedures can have several. Additionally, unlike stored procedures, views do not accept any parameters. The fact that views permit specific operations, such as insert, update, and delete, that procedures do not is another important benefit of views over processes.

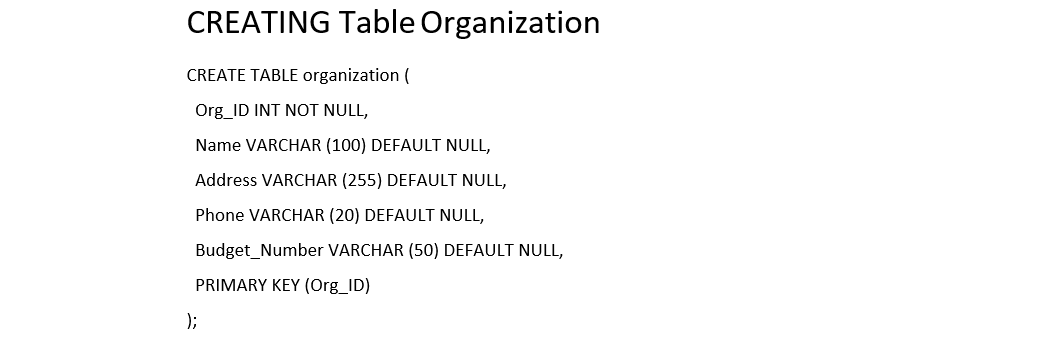
**VIEWS FOR person**

View older\_than\_1995 and

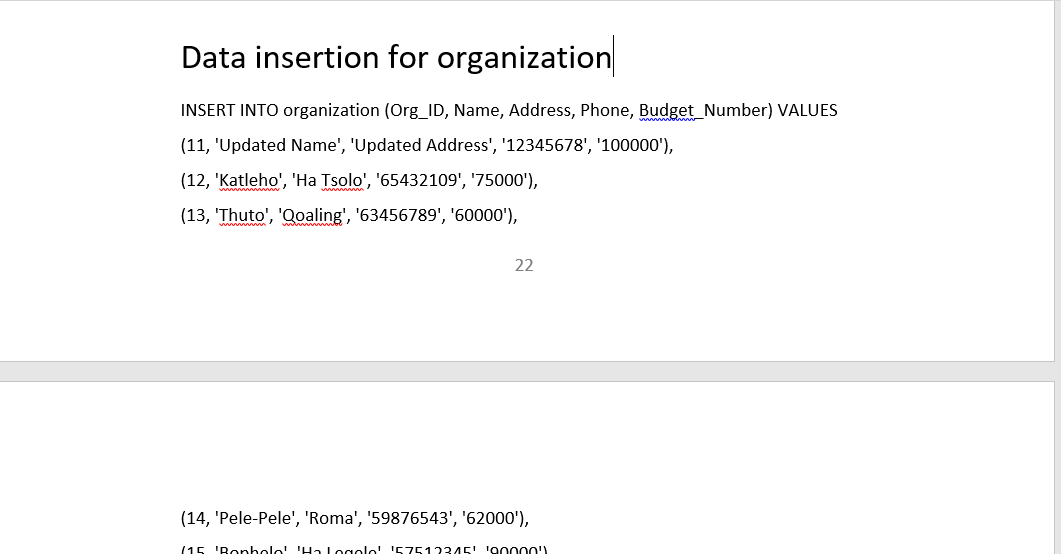
View hlotse\_maputsoe\_residents

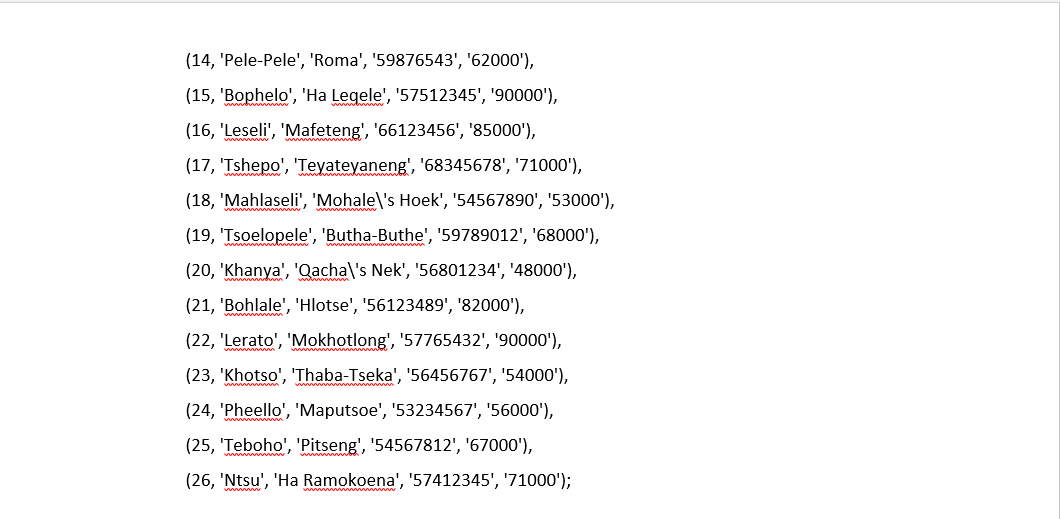


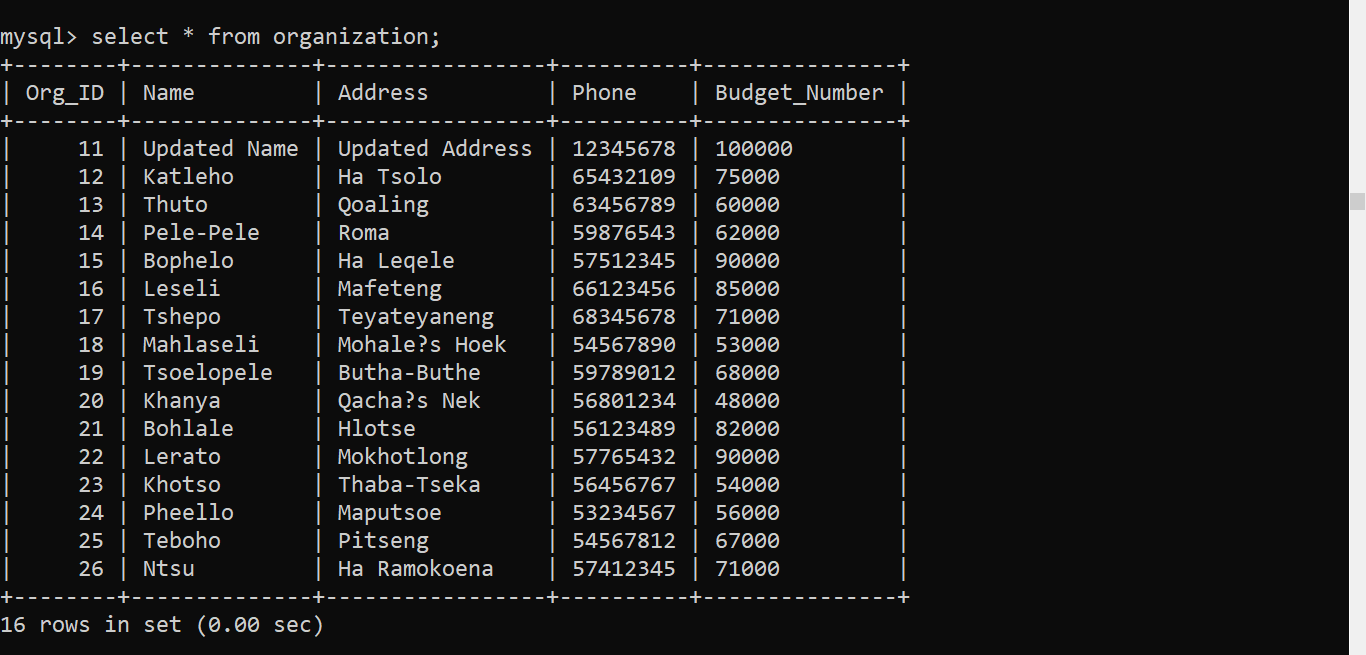
CREATING Table Organization



Data insertion for organization



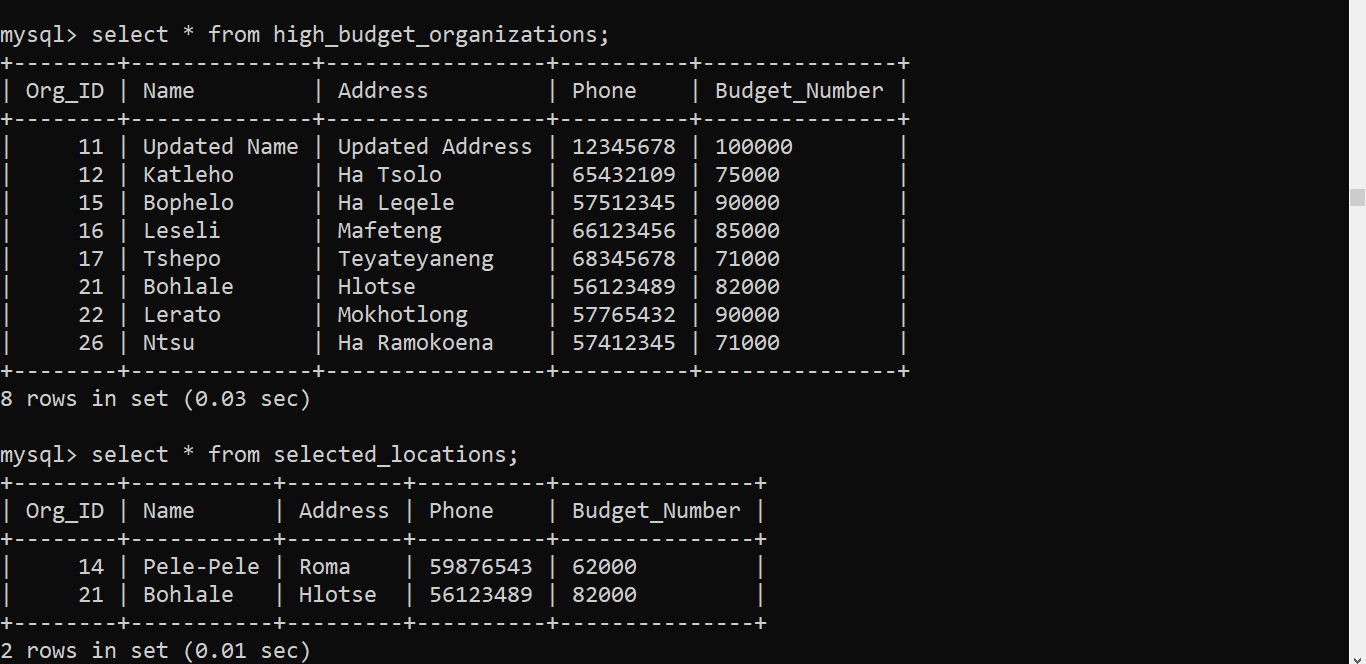




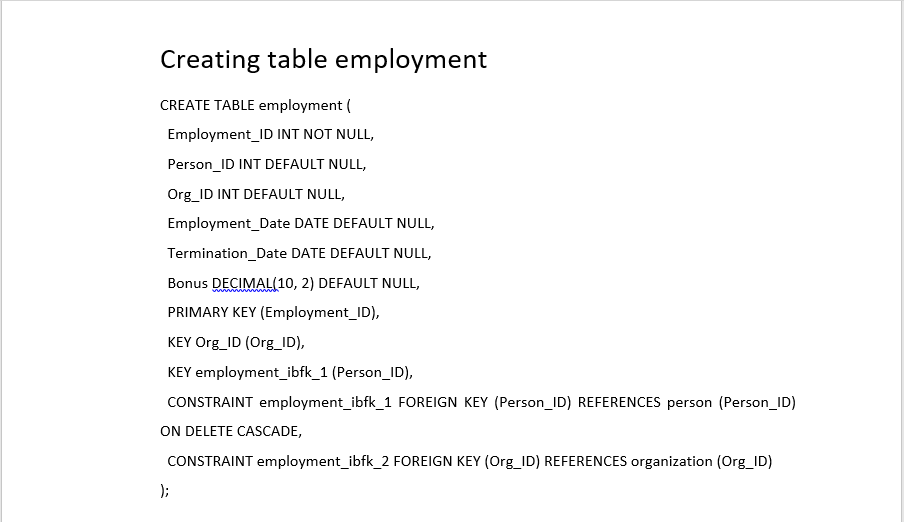
**CREATING VIEWs for table organization**

View where budget is greater than 70000, and

View where -> where Address IN ('Roma', 'Hlotse');

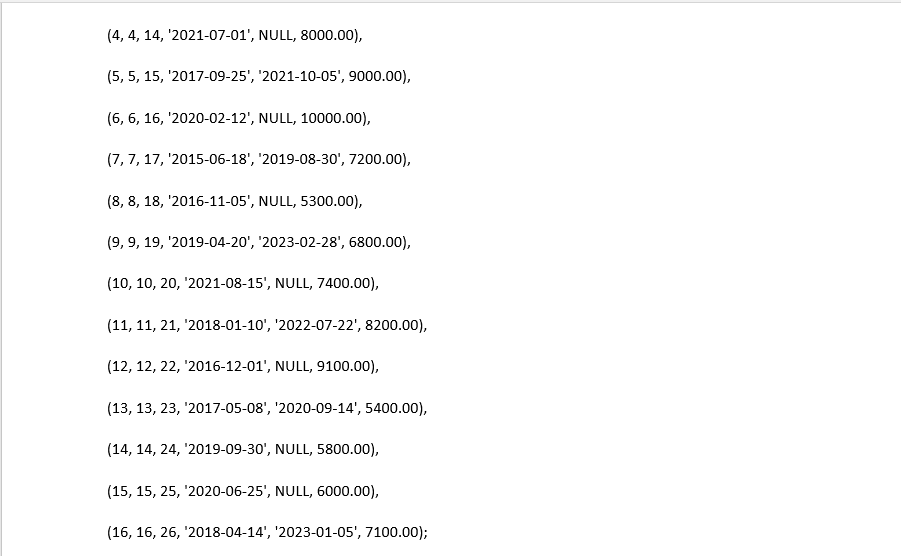


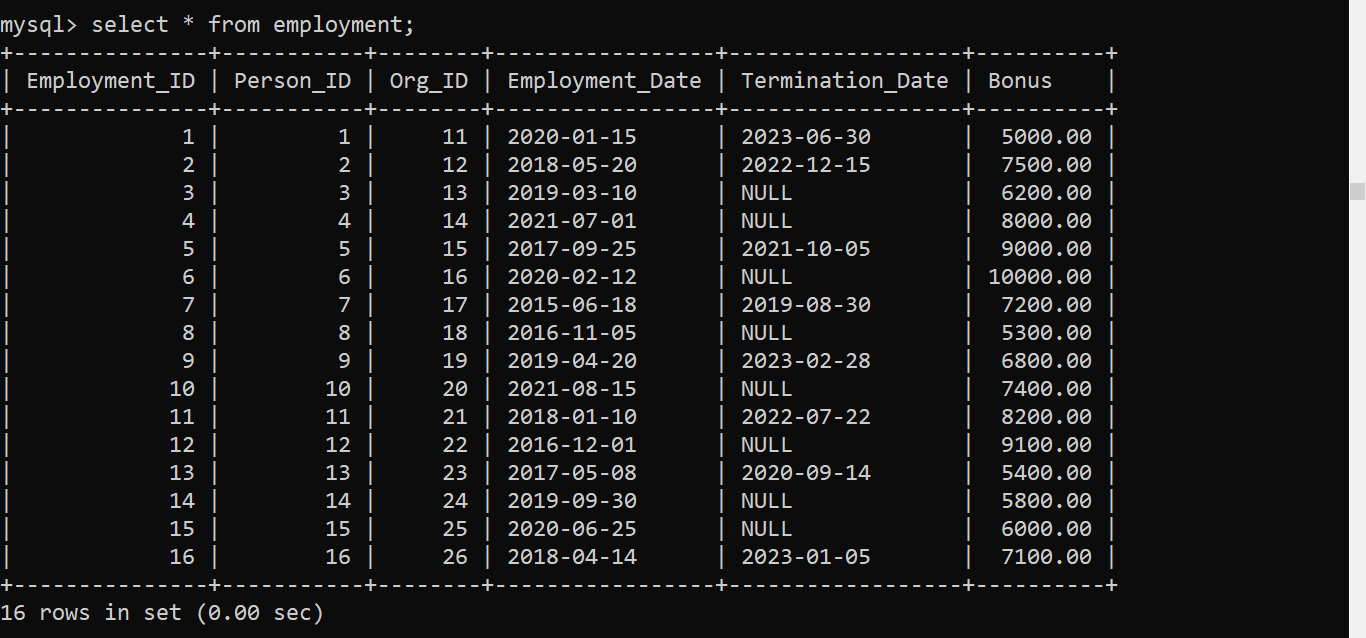
Creating table employment



Data insertion for employment



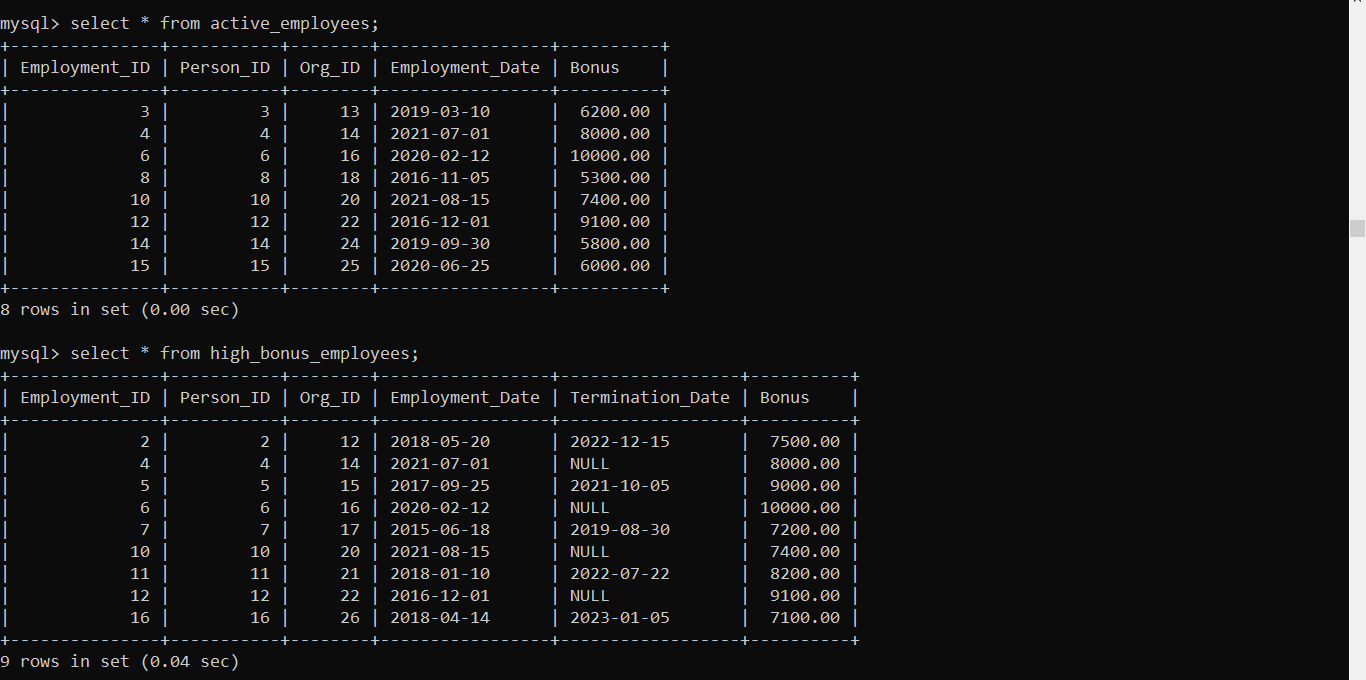




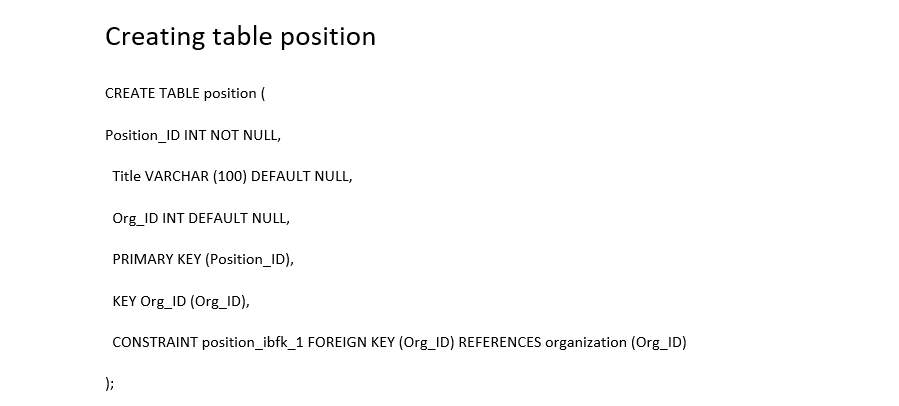
**CREATING VIEWS FOR TABLE employment**

View active\_employees and,

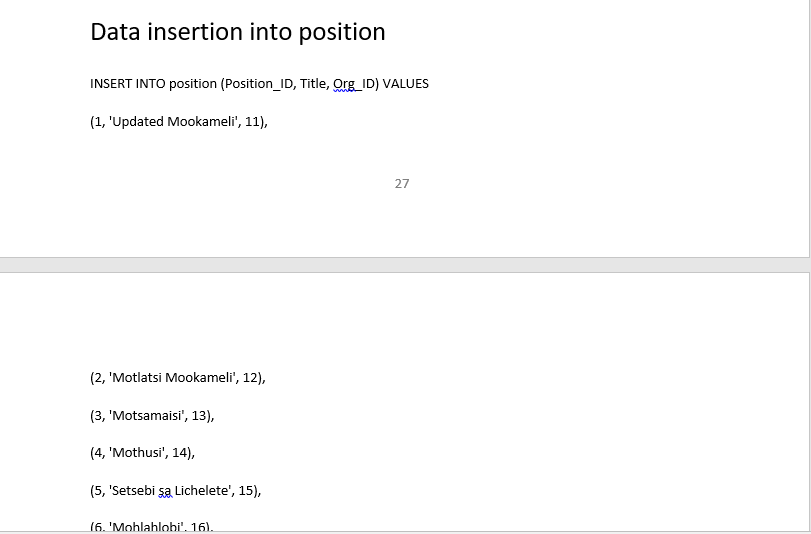
View High\_bonus\_employees

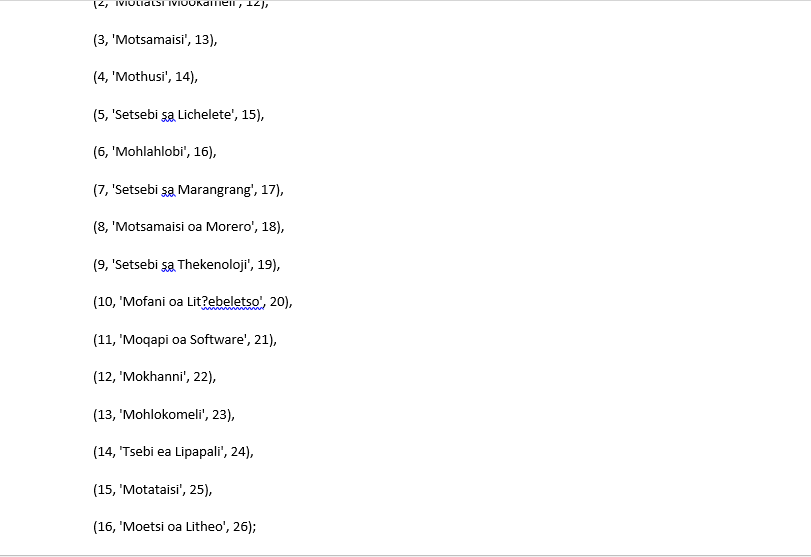


**Creating table position**

****

Data insertion into position

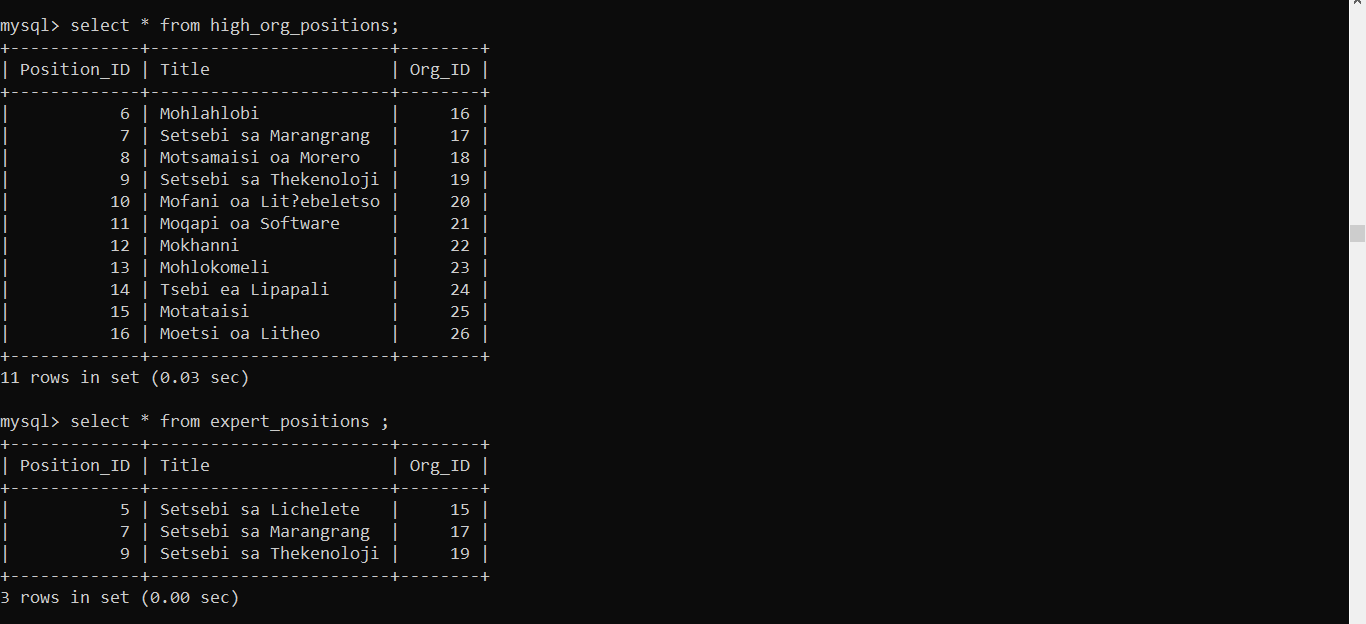




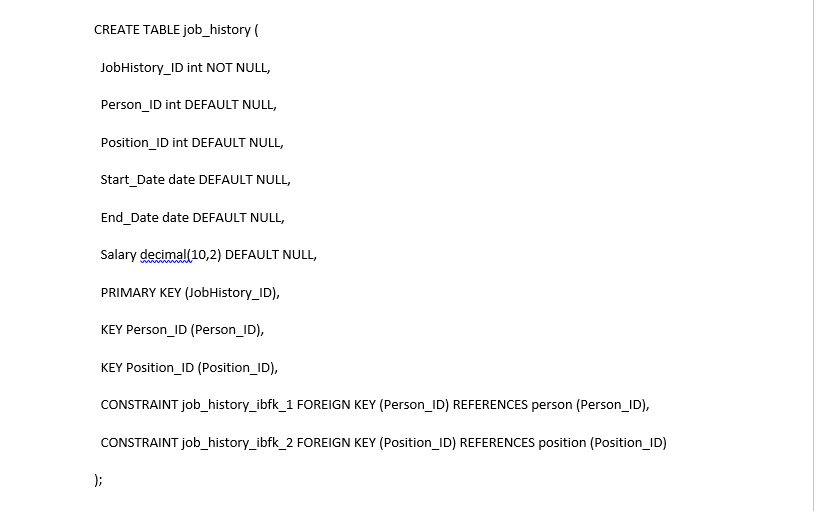
CREATING VIEWS FOR TABLE POSITION

View high\_org\_positions where Org ID is greater than 15

VIEW expert\_positions where Title like '%Setsebi%'

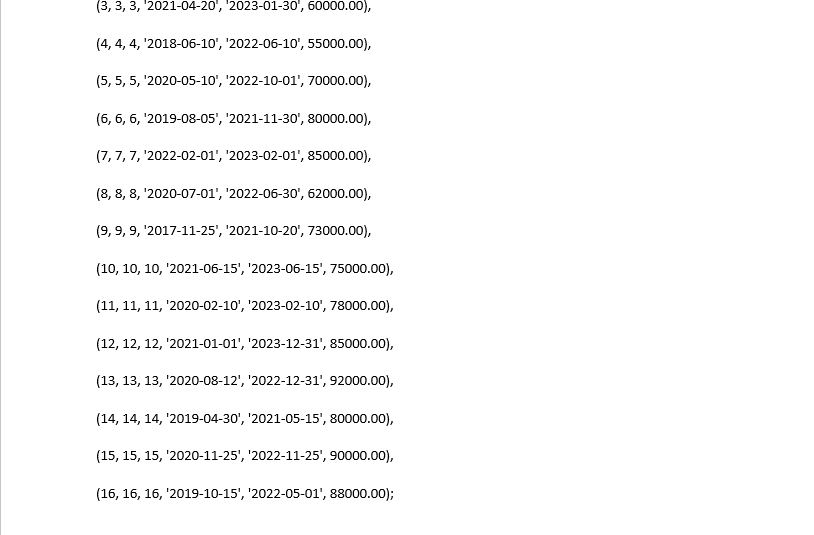


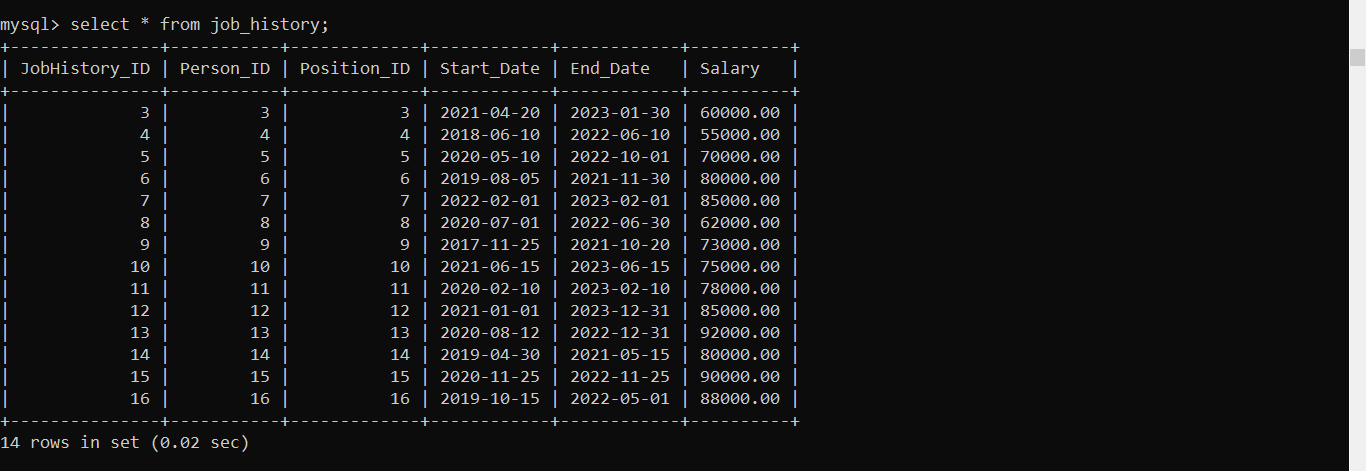
Creation of table job\_history



Data insertion into job\_history



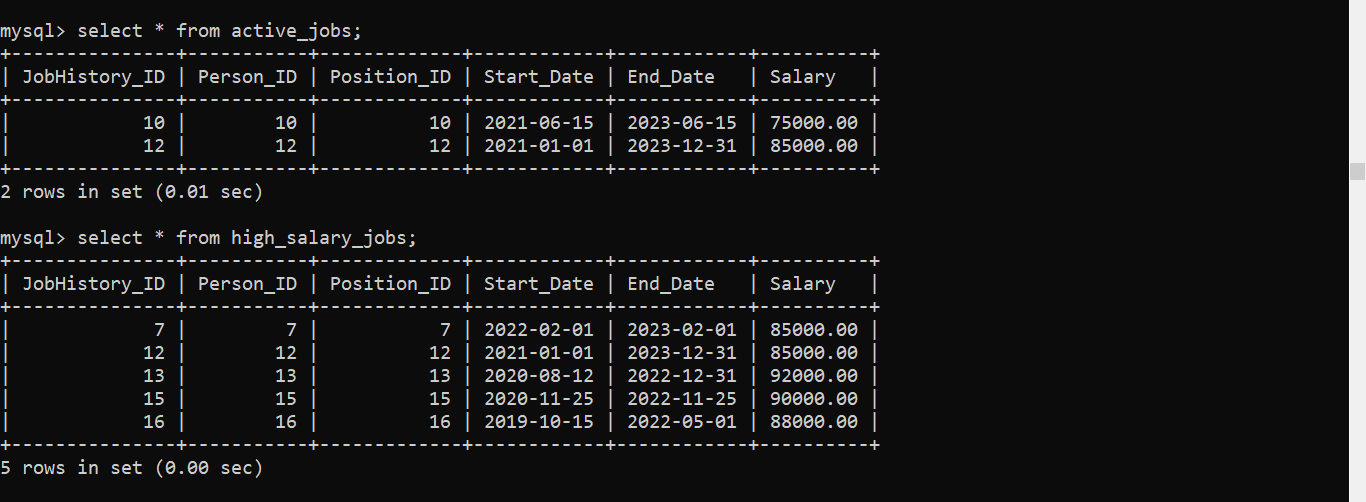




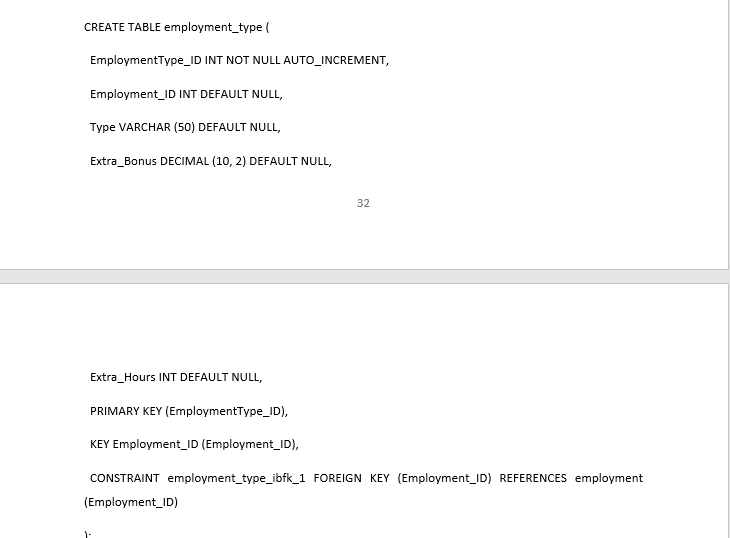
**CREATING VIEWS FOR TABLE job\_history**

View active\_jobs where End\_Date is NULL

VIEW high\_salary\_jobs where salary is greater than 80000

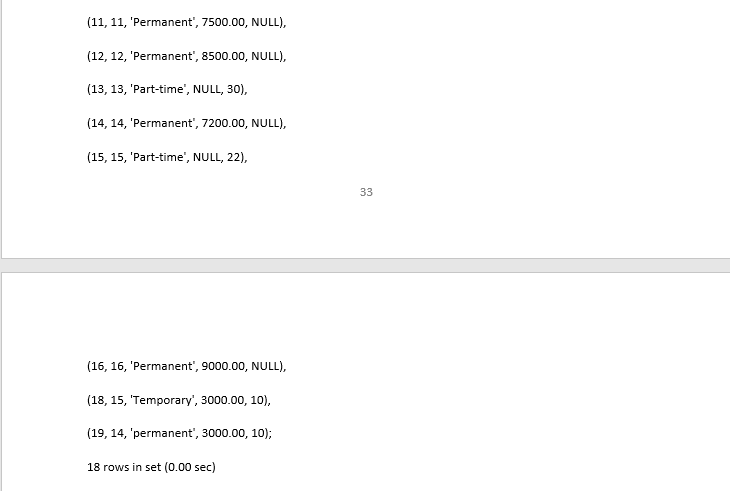


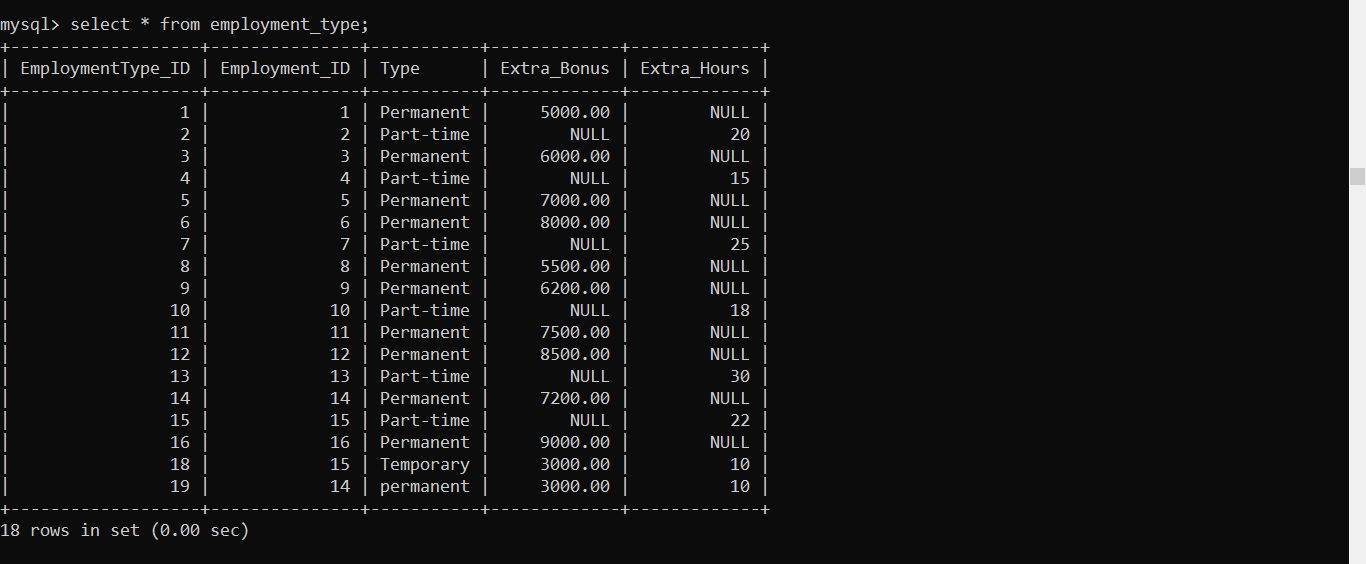
Creating table employment\_type



Data insertion into employment\_type



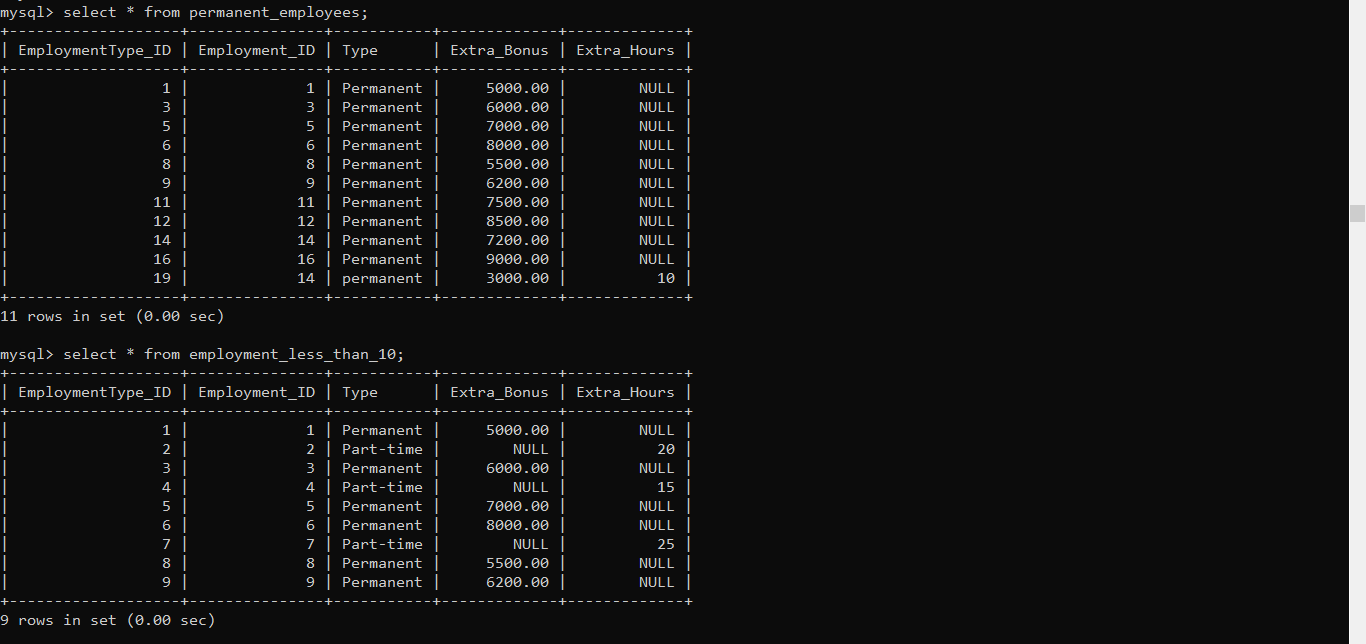




**CREATING VIEWS FOR TABLE employment\_type**

View permanent\_employees where type is permanent

View employmentID\_less\_than\_10 where employment\_type is less than 10

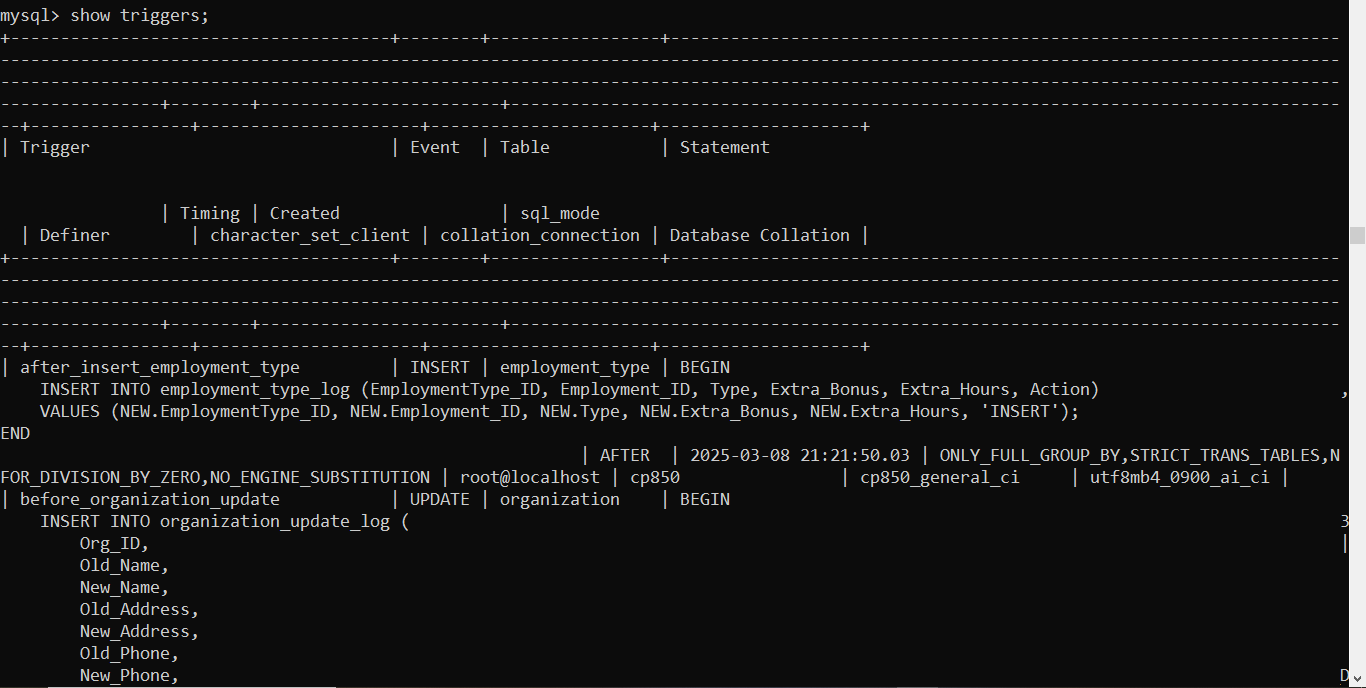


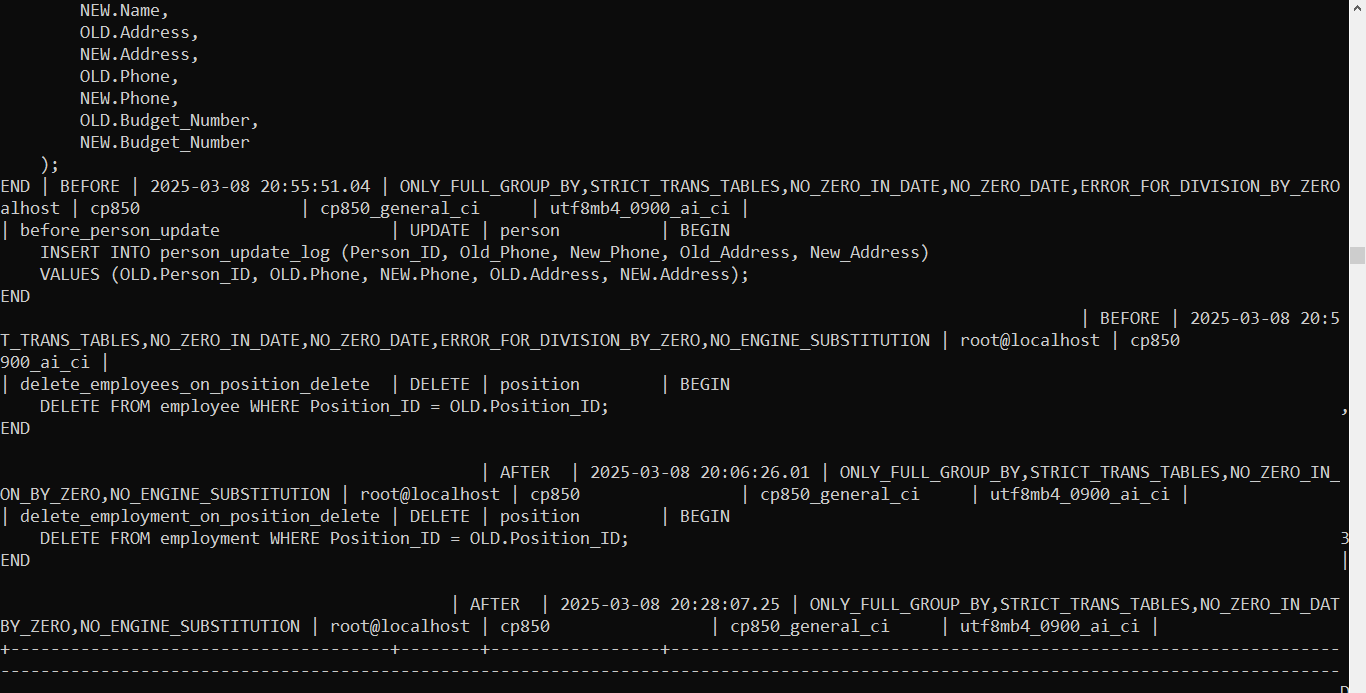
**TRIGGERS**

When a certain event takes place, a block of procedural code known as a SQL Server trigger is triggered. According to Kamlesh.S (2023), without the application or user having to explicitly intervene, triggers are used to automate specific tasks, enforce business rules, and preserve data integrity. It is possible to specify that triggers will run either before (BEFORE TRIGGER), after (AFTER TRIGGER), or in place of (INSTEAD OF TRIGGER) the data change action.

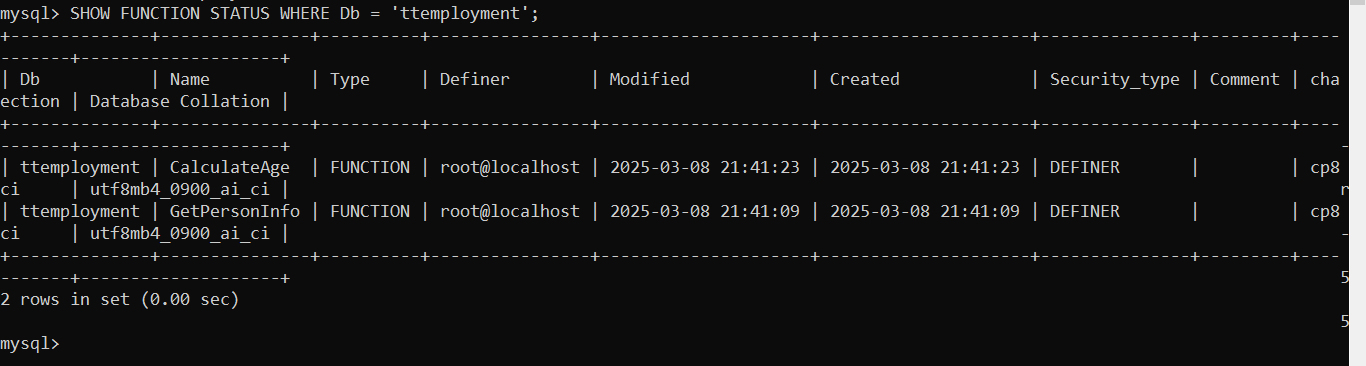
DML triggers that record events, as when a user INSERTS, UPDATES, or DELETES a row in a table that is issued in a SQL query or stored procedure, are the most widely used triggers. When a user connects to a database, SQL triggers can also be used to log (insert into a log table). Among other things, SQL triggers can log when a user or event makes changes to a table.

**Triggers**

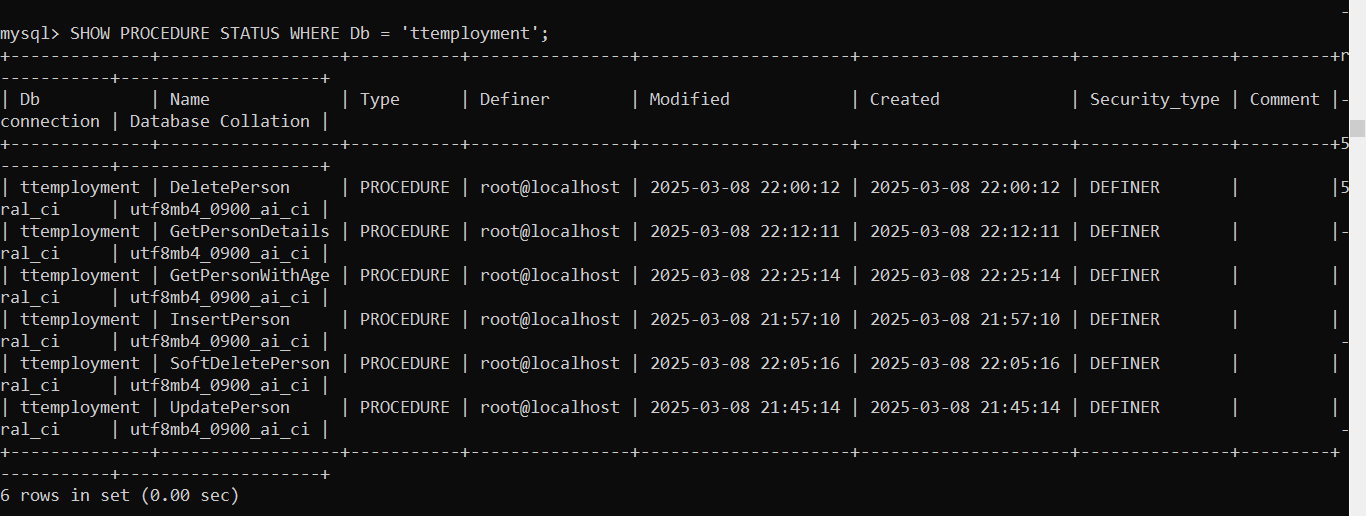
****

****

**FUNCTIONS**

****

**PROCEDURES**

****

**CREATING USERS AND GRANTING PRIVILEGE**

