Contents

[Abstract 2](#_Toc194573727)

[Chapter 1: Introduction 2](#_Toc194573728)

[1.1 Problem Statement 2](#_Toc194573729)

[1.2 Problem Solving 3](#_Toc194573730)

[1.3 Objective 3](#_Toc194573731)

[1.3 Scope & Constraint 3](#_Toc194573732)

[Chapter 2: Reviewing the Literature 4](#_Toc194573733)

[2.3 Findings & Discussion 4](#_Toc194573734)

[Chapter3: Methodology 4](#_Toc194573735)

[3.2 System Design 5](#_Toc194573736)

[3.2.1 Architectural design 5](#_Toc194573737)

[3.3 System Implementation / Prototyping 6](#_Toc194573738)

[3.4Testing 6](#_Toc194573739)

[Chapter 4: System Initiation and Planning 6](#_Toc194573740)

[4.1 Assessing Project Feasibility 6](#_Toc194573741)

[4.2 Project Plan 7](#_Toc194573742)

[Chapter 5: System Analysis 7](#_Toc194573743)

[5.1 Determining System Requirements 7](#_Toc194573744)

[5.2 Structuring System Requirements 8](#_Toc194573745)

[Chapter 6: Conclusion 8](#_Toc194573746)

[6.1 Advantages of the System 8](#_Toc194573747)

[6.3 Potential Benefit 8](#_Toc194573748)

[6.4 Conclusion 9](#_Toc194573749)

[what is an Entity Relationship Diagram(ERD)? 9](#_Toc194573750)

[ER/EER Diagram. 10](#_Toc194573751)

[Mapping 11](#_Toc194573752)

[What is Mapping? 11](#_Toc194573753)

Abstract  
The Employee and Position Management System is developed to oversee employment and position-related records for individuals and organizations. It stores essential information such as ID, name, address, phone number, and date of birth for individuals, while maintaining the budget number for organizations. Additionally, the system monitors employment details, including the start and end dates of employment, as well as any bonuses associated with specific positions. It also records position-related data, such as job title, commencement date, termination date, and salary. Designed to support both permanent and part-time employees, the system accounts for their respective benefits and additional working hours. This report details the methodology applied in the system’s design and implementation.

## Chapter 1: Introduction

This report aims to outline the design of a system for managing employment and position-related data for individuals and organizations. The system enables the recording and monitoring of employee details, job positions, and organizational information. It is designed for organizations that hire individuals and require a structured approach to maintaining comprehensive records of employment history and job roles.

### Problem Statement

The management of employee data in organizations can be a complex and challenging task, especially when handling multiple employees, organizations, and positions. It requires precise tracking of employment details, position history, and personal information for each employee and organization. Errors or inaccuracies in this data can result in payroll discrepancies, benefits miscalculations, and legal compliance issues. Therefore, implementing an efficient system to manage this data is crucial. Accurate record-keeping ensures timely salary payments, proper allocation of benefits, and compliance with employment regulations. However, managing employment and position details for both individuals and organizations can be time-consuming and prone to human errors (Smith & Johnson, 2015).

### Problem Solving

The proposed system aims to provide a solution to the problem of managing employee data in organizations. It will feature a centralized database to store information about employees, organizations, positions, and employment details. The system will facilitate easy data input, retrieval, and modification, ensuring accuracy and completeness (Brown D & Williams, R, 2020).

### 1.3 Objective

The objective of this project is to develop a system that can efficiently manage employee data for organizations. The system should be user-friendly and easily accessible to authorized personnel. It should also be capable of generating reports on employment, position history, and personal information for employees and organizations. The system will help automate various employment-related activities, improve the accuracy and timeliness of employee-related information, and improve overall organizational efficiency. The system aims to improve the accuracy of data and reduce the time required to manage employee and position details. (La exatlly 2015).

### Scope & Constraint

The system will focus on managing employee data for organizations, including personal information, employment history, and position history. It will also allow for the tracking of bonuses, credits, and extra hours for employees. However, the system will not handle payroll or benefit calculations, as these are typically managed by separate systems. The system will be designed for use by authorized personnel within an organization and will comply with applicable data privacy laws and regulations.

A key constraint of the system is that it will use MySQL as its database (Oracle Corporation, 2022).

## Chapter 2: Reviewing the Literature

Research by (Kwon, H., Lee, J. & Park, S., 2019) indicates that employee management systems play a crucial role in minimizing administrative workload, streamlining communication, and boosting overall organizational effectiveness. Similarly, Hong et al. (2019) (Hong, Y., Kim, D. & Cho, S., 2019) found that these systems contribute to higher levels of employee engagement, increased job satisfaction, and lower turnover rates.

The significance of data quality, accuracy, and security in employee management systems has been increasingly emphasized in recent studies. Poor data management, including errors or incomplete records, can lead to severe financial and legal repercussions for organizations. Consequently, ensuring data integrity through stringent validation and verification mechanisms is essential (Smith, 2013).

Further research has examined the structural design and implementation of employee data management systems, proposing various architectural models, database frameworks, and user interfaces that optimize employee data handling. Additionally, many modern systems incorporate cloud-based technology, allowing seamless access and collaboration across multiple departments and organizations (Brown D & Williams, R, 2020).

### 2.3 Findings & Discussion

Chapter3: Methodology

Existing research indicates that employee management systems serve as valuable tools for enhancing organizational efficiency, fostering employee engagement, and increasing job satisfaction. Implementing such a system can streamline various employment-related processes, ensure the timely and accurate management of employee information, and contribute to overall operational effectiveness. To maximize its benefits, the system should prioritize data integrity, security, and accuracy while maintaining a user-friendly interface for authorized users (Johnson & Miller, 2010) (Johnson, R. & Miller, T, 2010)  
3.1 Requirement Analysis

The system requirements will be established by analyzing the needs of organizations in managing employee data. This process will include discussions with human resources professionals and other relevant stakeholders to pinpoint the critical features and functionalities required. These requirements will be prioritized and outlined in a formal requirements specification document (Anderson & Thompson,, 2021).

### 3.2 System Design

The system will adopt a modular design, featuring distinct modules for employee data management, organization management, and position management. It will utilize a centralized database to facilitate easy access and retrieval of data. To maintain high standards of data quality, accuracy, and security, the system will incorporate thorough validation and verification processes (Brown, D & Clark, 2020).

### 3.2.1 Architectural design

The system architecture will be designed for scalability, enabling seamless expansion as organizations grow and evolve. It will be based on a cloud infrastructure and will be organized into four key layers: the presentation layer, the application layer, the business logic layer, and the data access layer. The presentation layer will include the graphical design of the software as well as the code responsible for managing user interaction, with only UI-specific logic found within this layer. The business logic layer will manage all business logic, validations, and processes related to business requirements. The data access layer will handle interactions with the database, also known as the persistence layer. Finally, the database layer will serve as the real data store for the software, consisting of the code to access and manage the database behind the software product, as well as the underlying database technology (Johnson & Lee,, 2022).

### 3.3 System Implementation / Prototyping

The system was developed using the .NET framework, with MySQL serving as the database solution. The development process followed agile methodologies, with continuous feedback from stakeholders to ensure that the system aligned with their requirements (Williams. & Davis, 2022)

3.4Testing  
As noted by (Shawkat, 2021), the system underwent both functional and non-functional testing. Functional testing focused on evaluating the core functionalities of the system, while non-functional testing assessed the system's performance, security, and usability. Usability testing, in particular, concentrated on evaluating how easy the system is for users to navigate, its flexibility in handling controls, and its ability to meet its intended objectives.

## Chapter 4: System Initiation and Planning

### 4.1 Assessing Project Feasibility

The initial step in any database project is to evaluate its feasibility. For this project, we are designing a database to manage the employment and position history of individuals across various organizations. This database must be capable of handling large volumes of data while remaining adaptable to changes in the employment and position histories of individuals over time. Ensuring the security of sensitive information is also a priority. To assess the feasibility, we need to evaluate several factors: technical feasibility, economic feasibility, and organizational feasibility.

Technical feasibility involves determining whether the project can be successfully implemented using existing technologies. In this case, we must consider the database management system and any other software or hardware requirements. Economic feasibility refers to the costs associated with the project and whether they are justified by the anticipated benefits. This includes the cost of developing, implementing, and maintaining the database. Organizational feasibility looks at whether the organization can effectively implement and use the system, considering the skills of the staff who will operate it and any training or support required.

Once we have assessed the feasibility, we can proceed with the system analysis phase (Jackson & Taylor, 2010).

### 4.2 Project Plan

According to (Peter, 2023), the project plan provides a comprehensive outline of the steps necessary for the successful completion of the project. It includes a detailed timeline, budget, and mechanisms for project monitoring and evaluation.

The first step in creating the project plan is to define the scope of the project. For this project, the scope involves designing and developing a system to manage employment records, positions, and salaries. Once the scope is determined, the next step is to create a detailed timeline, which should include specific milestones and deadlines for each project phase, such as design, development, testing, and deployment.

The project budget will outline the resources required, including personnel, hardware, software, and other associated costs. Additionally, the project monitoring and evaluation section will describe how the project's progress will be tracked and assessed to ensure it remains on schedule and meets the defined objectives.

## Chapter 5: System Analysis

### 5.1 Determining System Requirements

During the system analysis phase, we will identify the specific requirements for the database system. These requirements will be shaped by the needs of the organizations and individuals who will be using the system. To establish the system requirements, we will conduct a comprehensive analysis of the existing employment and position management systems in use within the organizations. Additionally, we will interview key stakeholders to gather their perspectives on the functionalities the new system should offer.

The system requirements will specify the types of data to be stored in the database, the tables and fields to be included, and any data constraints. Furthermore, we will assess user requirements, including the types of reports needed and the essential user interface components for effective operation (Li & Wang, 1996).

### 5.2 Structuring System Requirements

## Chapter 6: Conclusion

After determining the system requirements, we will organize them in a clear and structured manner. This will involve creating a data model, which serves as a visual representation of the database structure. The data model will illustrate the tables and their interrelationships, as well as any data constraints. Additionally, we will develop a detailed schema that specifies the fields in each table, along with their respective data types and constraints. Once the data model and schema are finalized, we can proceed to the system implementation phase (Johnson, R. & Miller, T, 2010).

### 6.1 Advantages of the System

The employment and position management database system we have developed provides several advantages over current systems. By utilizing a centralized database for all employment and position history data, we can minimize duplication and improve data accuracy. This enables organizations to make better-informed decisions regarding hiring and promotions.

Additionally, the system enhances security by offering appropriate access controls to protect sensitive information. It is also sufficiently flexible to accommodate changes in employment and position history over time, ensuring that the data remains accurate and up-to-date. Moreover, the system supports the tracking of bonuses and extra hours for part-time employees (Roberts & Harrison, 2022)

### 6.3 Potential Benefit

The system offers significant advantages to organizations by streamlining the management of employment and position data. Additionally, it benefits employees by providing a centralized platform where they can easily access and view their employment history and job responsibilities (KPMG., 2022).

### 6.4 Conclusion

The system design presented in this report offers a comprehensive solution for managing employment and position information. It is designed to be scalable, flexible, and user-friendly. The system has the potential to benefit both organizations and employees by providing a centralized platform for managing employment and position data efficiently.

# what is an Entity Relationship Diagram(ERD)?

is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within the system.

ER diagrams are most often used to design or debug relational database in the fields of software engineering, business information systems,

education and research. Also known as ERDs or ER models, they use a defined set of symbols such as rectangles, diamonds ovals and connecting

lines to depict the interconnectedness of entities, relationships and their attributes says (Anon., n.d.)

|  |
| --- |
| Employment |
| PK: EmploymentID  FK: PersonID  FK: OrganisationID  EmploymentDate  TerminationDate  Bonus |

## ER/EER Diagram.

|  |
| --- |
| Organisation |
| Address  Phone  BudgetNumber |

|  |
| --- |
| Person |
| PK: ID  Name  Address  Phone  Birthdate |

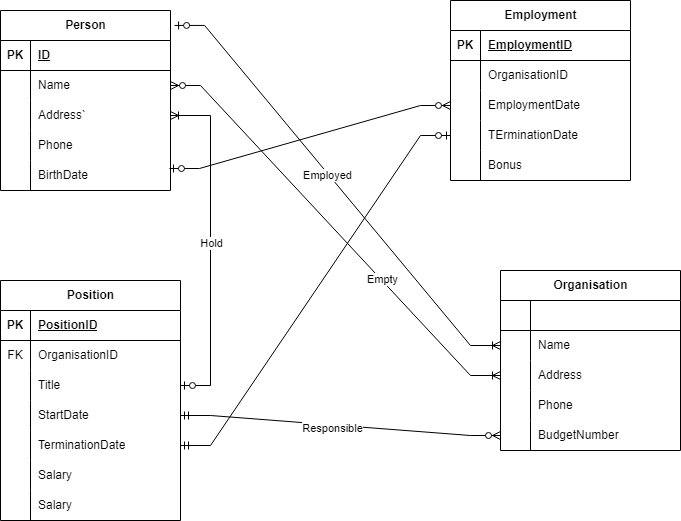
|  |
| --- |
| Person |
| PK: PositionID  FK: OrganisationID  Title  StartDate  TerminationDate  Salary |

|

Mapping.

### What is Mapping?

Mapping in the context of database design refers to the process of converting a conceptual schema into a logical or physical schema. This includes translating entity-relationship (ER) diagrams into a structured database schema comprised of tables that can be implemented in a database management system like SQL say by (Connolly, 2005)



CREATE TABLE Person (

ID NUMBER PRIMARY KEY,

Name VARCHAR2(50),

Address VARCHAR2(100),

Phone VARCHAR2(20),

Birthdate DATE

);

CREATE TABLE Organisation (

ID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Address VARCHAR2(100),

Phone VARCHAR2(20),

BudgetNumber NUMBER

);

CREATE TABLE Employment (

PersonID NUMBER,

OrganizationID NUMBER,

EmploymentDate DATE,

TerminationDate DATE,

Bonus NUMBER,

FOREIGN KEY (PersonID) REFERENCES Person(ID),

FOREIGN KEY (OrganizationID) REFERENCES Organization(ID)

);

CREATE TABLE Position (

Title VARCHAR2(100),

OrganizationID INT,

StartDate DATE,

TerminationDate DATE,

Salary NUMBER,

FOREIGN KEY (OrganizationID) REFERENCES Organization(ID)

);

SQL> CREATE TABLE Person (ID NUMBER PRIMARY KEY NOT NULL NUMBER, Name VARCHAR2(50), Address VARCHAR2(100), Phone\_Number VARCHAR(20), Birth\_Date DATE);

Table created.

SQL> desc person

Name Null? Type

----------------------------------------- -------- ----------------------------

ID NOT NULL NUMBER

NAME VARCHAR2(50)

ADDRESS VARCHAR2(100)

PHONE\_NUMBER VARCHAR2(20)

BIRTH\_DATE DATE

SQL> INSERT INTO Person

2 VALUES(102, ' Naleli Sello', 'Maseru', '5786745', TO\_DATE('2000-08-10', 'YYYY-MM-DD'));

1 row created.

SQL> INSERT INTO Person

2 VALUES(103, 'Paseka Ramafatle', 'Leribe', '63298110', TO\_DATE('1999-10-30', 'YYYY-MM-DD'));

1 row created.

SQL> INSERT INTO Person

2 VALUES(104, 'Thabo Ranaka', 'Quthing', '57990738', TO\_DATE('1995-03-22', 'YYYY-MM-DD'));

1 row created.

SQL> INSERT INTO Person

2 VALUES(105, 'Lereko Theko', 'Roma', '62899654 ', TO\_DATE('1999-05-22', 'YYYY-MM-DD'));

1 row created.

SQL> Select\* from Person;

ID NAME

---------- --------------------------------------------------

ADDRESS

--------------------------------------------------------------------------------

PHONE\_NUMBER BIRTH\_DAT

-------------------- ---------

102 Naleli Sello

Maseru

62789878 10-AUG-00

103 Paseka Ramafatle

Leribe

63298110 30-OCT-99

ID NAME

---------- --------------------------------------------------

ADDRESS

--------------------------------------------------------------------------------

PHONE\_NUMBER BIRTH\_DAT

-------------------- ---------

104 Thabo Ranaka

Quthing

57990738 22-MAR-95

105 Lereko Theko

Roma

ID NAME

---------- --------------------------------------------------

ADDRESS

--------------------------------------------------------------------------------

PHONE\_NUMBER BIRTH\_DAT

-------------------- ---------

62899654 22-MAY-94

4 rows selected.

SQL> CREATE TABLE Employment (Person\_ID NUMBER, Organisation\_ID NUMBER, Employment\_Date DATE, Termination\_Date DATE, Bonus NUMBER, FOREIGN KEY (Person\_ID) REFERENCES Person(ID), FOREIGN KEY (Organisation\_ID) REFERENCES Organisation(ID));

SQL> desc Employment

Name Null? Type

----------------------------------------- -------- ----------------------------

PERSON\_ID NUMBER

ORGANISATION\_ID NUMBER

EMPLOYMENT\_DATE DATE

TERMINATION\_DATE DATE

BONUS NUMBER

SQL> INSERT INTO Employment

2 VALUES (102, 002, TO\_DATE('2022-07-20', 'YYYY-MM-DD'),NULL, 4000);

1 row created.

SQL> INSERT INTO Employment

2 VALUES (103, 003, TO\_DATE('2021-07-01', 'YYYY-MM-DD'),NULL, 3000);

1 row created.

SQL> INSERT INTO Employment

2 VALUES (104, 004, TO\_DATE('2011-10-18', 'YYYY-MM-DD'),NULL, 10000);

1 row created.

SQL> select\* from Employment;

PERSON\_ID ORGANISATION\_ID EMPLOYMEN TERMINATI BONUS

---------- --------------- --------- --------- ----------

102 2 20-JUL-22 4000

103 3 01-JUL-21 3000

104 4 11-OCT-18 10000

CREATE TABLE Organisation (ID NUMBER PRIMARY KEY, Name VARCHAR2(100), Address VARCHAR2(100), Phone\_Number VARCHAR(20), Budget\_Number NUMBER);

SQL> desc Organisation

Name Null? Type

----------------------------------------- -------- ----------------------------

ID NOT NULL NUMBER

NAME VARCHAR2(100)

ADDRESS VARCHAR2(100)

PHONE\_NUMBER VARCHAR2(20)

BUDGET\_NUMBER NUMBER

SQL> INSERT INTO Organisation

2 VALUES (002, 'ABC Foundation', 'Kings Way Road', '22334567', 15000);

1 row created.

SQL> INSERT INTO Organisation

2 VALUES (003, 'Masako Connections', 'NRH Mall Room 2', '22339403' , 25000);

1 row created.

SQL> INSERT INTO Organisation

2 VALUES (004, 'Leseko Savings and Credit', 'Platium Room 7', '67392920', 35000);

1 row created.

SQL> INSERT INTO Organisation

2 VALUES (005, 'Lesego holdings', 'Maseru Mall', '22345678', 35000);

1 row created.

SQL> SELECT\* FROM Organisation;

ID

----------

NAME

--------------------------------------------------------------------------------

ADDRESS

--------------------------------------------------------------------------------

PHONE\_NUMBER BUDGET\_NUMBER

-------------------- -------------

2

ABC Foundation

Kings Way Road

22334567 15000

ID

----------

NAME

--------------------------------------------------------------------------------

ADDRESS

--------------------------------------------------------------------------------

PHONE\_NUMBER BUDGET\_NUMBER

-------------------- -------------

3

Masako Connections

NRH Mall Room 2

22339403 25000

ID

----------

NAME

--------------------------------------------------------------------------------

ADDRESS

--------------------------------------------------------------------------------

PHONE\_NUMBER BUDGET\_NUMBER

-------------------- -------------

4

Leseko Savings and Credit

Platium Room 7

67392920 35000

ID

----------

NAME

--------------------------------------------------------------------------------

ADDRESS

--------------------------------------------------------------------------------

PHONE\_NUMBER BUDGET\_NUMBER

-------------------- -------------

5

Lesego holdings

Maseru Mall

22345678 35000

SQL> CREATE TABLE Position (Title VARCHAR2(50), Organisation\_ID NUMBER, Start\_Date DATE, Termination\_Date DATE, Salary NUMBER, FOREIGN KEY (Organisation\_ID) REFERENCES Organisation(ID));

SQL> desc Position;

Name Null? Type

----------------------------------------- -------- ----------------------------

TITLE VARCHAR2(50)

ORGANISATION\_ID NUMBER

START\_DATE DATE

TERMINATION\_DATE DATE

SALARY NUMBER

SQL> INSERT INTO Position

2 VALUES ('Sales Representative ', 002, TO\_DATE('2022-10-30', 'YYYY-MM-DD'), NULL, 45000);

1 row created.

SQL> INSERT INTO Position

2 VALUES ('Manager', 003, TO\_DATE('2029-02-02', 'YYYY-MM-DD'), NULL, 55000);

1 row created.

SQL> INSERT INTO Position

2 VALUES ('Senior Manager', 004, TO\_DATE('2000-01-01', 'YYYY-MM-DD'), NULL, 55000);

1 row created.

SQL> INSERT INTO Position

2 VALUES ('Accountant', 005, TO\_DATE('2023-03-01', 'YYYY-MM-DD'), NULL, 6000);

1 row created.

SQL> select\* from Position;

TITLE ORGANISATION\_ID START\_DAT

-------------------------------------------------- --------------- ---------

TERMINATI SALARY

--------- ----------

Sales Representative 2 30-OCT-22

45000

Manager 3 02-FEB-29

55000

Senior Manager 4 01-JAN-00

70000

TITLE ORGANISATION\_ID START\_DAT

-------------------------------------------------- --------------- ---------

TERMINATI SALARY

--------- ----------

Accountant 5 01-MAR-23

6000

SQL> SELECT TEXT

2 FROM USER\_VIEWS

3 WHERE VIEW\_NAME = 'PERSON\_DETAILS';

TEXT

--------------------------------------------------------------------------------

SELECT ID, Name, Address, Phone\_Number, Birth\_Date FROM Person

SQL> SELECT TEXT

2 FROM USER\_VIEWS

3 WHERE VIEW\_NAME = 'EMPLOYMENT\_DETAILS';

TEXT

--------------------------------------------------------------------------------

SELECT Person.Name, Organisation.Name AS Organisation\_Name, Employment.Employmen

SQL> SELECT ID, Name FROM Person

2 UNION

3 SELECT ID, Name FROM Organisation;

ID

----------

NAME

--------------------------------------------------------------------------------

102

Naleli

103

Paseka

104

Thabo

ID

----------

NAME

--------------------------------------------------------------------------------

2

ABC Foundation

3

Masako Connections

4

Leseko Savings and Credit

6 rows selected.

SQL> CREATE OR REPLACE PROCEDURE Update\_Employment\_Bonus(p\_person\_id NUMBER, p\_bonus NUMBER) AS

2 BEGIN

3 UPDATE Employment SET Bonus = p\_bonus WHERE Person\_ID = p\_person\_id;

4 COMMIT;

5 END Update\_Employment\_Bonus;

6 /

Procedure created.

SQL> BEGIN

2 Update\_Employment\_Bonus(104, 9000);

3 END;

4 /

PL/SQL procedure successfully completed.

SQL> EXEC Update\_Employment\_Bonus(104, 9000);

PL/SQL procedure successfully completed.

SQL> CREATE OR REPLACE PROCEDURE Update\_Phone\_Number (p\_person\_id IN NUMBER, p\_new\_phone IN VARCHAR2) IS

2 BEGIN

3 UPDATE Person

4 SET phone\_number = p\_new\_phone\_number

5 WHERE person\_id = p\_person\_id;

6 COMMIT;

7 DBMS\_OUTPUT.PUT.LINE('Phone number updated successfully.');

8 EXCEPTION

9 WHEN NO\_DATA\_FOUND THEN

10 DBMS\_OUTPUT.PUT.LINE('Person with ID' || p\_person\_id || 'not found.');

11 WHEN NO\_DATA\_FOUND THEN

12 DBMS\_OUTPUT.PUT.LINE('Error updating phone number');

13 END; Update\_Phone\_Number;

Procedure created.

SQL> BEGIN

2 Update\_Phone\_Number(102, '62789878');

3 END;

4 /

PL/SQL procedure successfully completed.

SQL> EXEC Update\_Employment\_Bonus(104, 9000);

PL/SQL procedure successfully completed.

SQL> CREATE OR REPLACE FUNCTION GetEmployeeCount(org\_id INT)

2 RETURN INT

3 AS

4 total\_employees INT;

5 BEGIN

6 SELECT COUNT(DISTINCT Person\_ID)

7 INTO total\_employees

8 FROM Employment

9 where org\_ID = ORG\_ID;

10 RETURN total\_employees;

11 END;

12 /

Function created.

SQL> select\* from Employment;

PERSON\_ID ORGANISATION\_ID EMPLOYMEN TERMINATI BONUS

---------- --------------- --------- --------- ----------

102 2 20-JUL-22 4000

103 3 01-JUL-21 3000

104 4 11-OCT-18 9000

SQL> CREATE OR REPLACE PROCEDURE Terminate\_Employee(employment\_id INT, term\_date DATE) AS

2 BEGIN

3 UPDATE Employment;

4 SET Termination\_Date = term\_date

5 WHERE Person\_ID = emp\_id;

6 COMMIT;

7 END;

8 /

Warning: Procedure created with compilation errors.

SQL> BEGIN

2 Terminate\_Employee(103, TO\_DATE('2021-07-01' YYYY-MM-DD'));

3 END;

4 /

ERROR:

ORA-01756: quoted string not properly terminated

SQL> BEGIN

2 Terminate\_Employee(103, TO\_DATE('2021-07-01' YYYY-MM-DD'));

3 DBMS\_OUTPUT.PUT\_LINE('Employee Terminated successfully.');

4 END;

5 /

ERROR:

ORA-01756: quoted string not properly terminated

SQL> EXEC Terminate\_Employee(103, TO\_DATE('2021-07-01' YYYY-MM-DD');

ERROR:

ORA-01756: quoted string not properly terminated

SQL> SELECT\* FROM Employment;

PERSON\_ID ORGANISATION\_ID EMPLOYMEN TERMINATI BONUS

---------- --------------- --------- --------- ----------

102 2 20-JUL-22 4000

103 3 01-JUL-21 3000

104 4 11-OCT-18 9000

SQL> CREATE USER c##Thabo IDENTIFIED BY "Thizozo101";

User created.

SQL> CREATE USER c##Naleli IDENTIFIED BY "Nelly007";

User created.

SQL> CREATE USER c##Paseka IDENTIFIED BY "Pascos567";

User created.

SQL> GRANT SELECT ON Person TO c##Thabo;

Grant succeeded.

SQL>

SQL> GRANT SELECT ON Person TO c##Naleli;

Grant succeeded.

SQL>

SQL> GRANT SELECT ON Person TO c##Paseka;

Grant succeeded.