### TT Holding Database Design Report

#### Table of Contents

1. **Abstract**
2. **Introduction**
   * 2.1 Problem Statement
   * 2.2 Problem Solving
   * 2.3 Objective
   * 2.4 Scope & Constraint
3. **Literature Review**
   * 3.1 Introduction
   * 3.2 Reviewing the Literature
   * 3.3 Findings & Discussion
4. **Methodology**
   * 4.1 Requirement Analysis
   * 4.2 System Design
     + 4.2.1 Architectural Design
     + 4.2.2 UML Diagrams
   * 4.3 System Implementation-Design / Prototyping
   * 4.4 Testing
5. **System Initiation and Planning**
   * 5.1 Assessing Project Feasibility
   * 5.2 Project Plan
6. **System Analysis**
   * 6.1 Determining System Requirements
   * 6.2 Structuring System Requirements
7. **Conclusion**
   * 7.1 Advantages of the System
   * 7.2 Future Enhancement of the System
   * 7.3 Potential Benefits

7.4 Conclusion

1. **References**
2. **-35 Appendices**
3. **Abstract**

The TT Holding Database Design project is about creating a system to manage employee and organizational data. This report explains how the system was designed, built, and tested. The database helps store information about employees, their jobs, and the organizations they work for. It also ensures that the data is safe, accurate, and easy to access. The system uses tools like SQL and normalization to make sure the database works well. This project shows how a good database can help organizations manage their data better and make smarter decisions. The system is designed to be scalable, meaning it can grow as the organization grows. It also includes features like access control to protect sensitive data. Overall, the project demonstrates the importance of using technology to improve how organizations handle information.

**2.Introduction**  
**2.1 Problem Statement**  
Many organizations struggle to manage employee and organizational data. Without a proper system, data can become messy, duplicated, or even lost. TT Holdings, a growing company, needed a better way to store and manage information about its employees, their jobs, and the organizations they work for. The old system was slow, unreliable, and made it hard to find important information. For example, when managers needed to find details about an employee's job history, they had to search through multiple files, which took a lot of time. This inefficiency caused delays in decision-making and created frustration among employees. Additionally, the lack of a centralized system made it difficult to ensure data accuracy and security.

**2.2 Problem Solving**  
To solve these problems, we created a new database system. This system uses a relational database model, which is a way of organizing data into tables. It also uses normalization, a process that reduces duplicate data and keeps everything consistent. The system is designed to be fast, secure, and easy to use. For example, employees can now quickly search for information using simple queries. Managers can generate reports in minutes instead of hours. The system also includes features like access control, which ensures that only authorized users can view or modify sensitive data. By automating many tasks, the system reduces the risk of errors and saves time for employees.

**2.3 Objective**  
The main goal of this project is to build a database system that can store and manage employee and organizational data. The system should make it easy to add, update, and find information. It should also keep the data safe and secure. For example, the system should allow HR managers to quickly add new employees, update their job details, and generate reports. It should also provide tools for tracking employee performance and managing budgets. Another objective is to ensure that the system is scalable, meaning it can handle more data as the organization grows. Finally, the system should be user-friendly, so employees can use it without needing extensive training.

**2.4 Scope and Constraint**  
The system will handle data about employees, their jobs, and the organizations they work for. It will also store information about salaries and budgets. However, there are some challenges. The project has limited time and resources. The system will need to be updated in the future to handle more data and new features. For example, the current version of the system does not include advanced analytics tools, but these could be added later. Another constraint is the need to ensure data security, especially when dealing with sensitive information like salaries and personal details. Despite these challenges, the system is designed to be flexible and adaptable, so it can meet the organization's needs both now and in the future.

**3. Literature Review**  
**3.Introduction**  
A good database system is important for any organization. It helps store and manage data efficiently. This section looks at what other experts have said about database design and how it can help organizations. For example, many experts agree that a well-designed database can improve decision-making by providing accurate and up-to-date information. It can also reduce costs by eliminating the need for manual data entry and storage. Additionally, a good database system can help organizations comply with regulations by ensuring that data is stored securely and can be easily audited.

**3.2 Reviewing the Literature**  
Experts agree that relational databases are the best way to store and manage data. They use tables to organize information, which makes it easy to find and update. Normalization is a process that helps reduce duplicate data and keeps everything consistent. For example, instead of storing the same information in multiple places, normalization ensures that each piece of data is stored in only one place. This reduces the risk of errors and makes the database more efficient. SQL is a language used to work with databases. It allows users to add, update, and find data quickly. For example, a manager can use SQL to generate a report showing all employees who have been with the company for more than five years. The literature also highlights the importance of access control, which ensures that only authorized users can view or modify sensitive data.

**3.3 Findings & Discussion**  
The research shows that a well-designed database can make a big difference. It can help organizations manage their data better, reduce errors, and save time. A good database system is also scalable, meaning it can grow as the organization grows. For example, if the organization hires more employees or opens new branches, the database can be expanded to handle the additional data. The research also emphasizes the importance of user training. Even the best database system will not be effective if employees do not know how to use it. Therefore, organizations should invest in training programs to ensure that employees can use the system effectively. Overall, the literature suggests that a well-designed database system is a key tool for any organization that wants to improve its operations and make better decisions.

**4.Methodology**  
**4.1 Requirement Analysis**  
The first step was to understand what the organization needed. We talked to employees and managers to find out what problems they were facing. We also looked at the old system to see what could be improved. Based on this, we created a list of requirements for the new system. For example, one requirement was that the system should allow employees to quickly search for information using simple queries. Another requirement was that the system should include tools for generating reports. We also identified the need for access control, so that only authorized users could view or modify sensitive data. Finally, we determined that the system should be scalable, so it could handle more data as the organization grew.

**4.2 System Design**

**4.2. 1Architectural Design**

The system is divided into three parts: the database, the application, and the user interface. The database stores the data, the application handles the logic, and the user interface lets people interact with the system. For example, when an employee searches for information, the user interface sends a request to the application, which then retrieves the data from the database. This three-tier architecture ensures that the system is modular and easy to maintain. It also makes it easier to add new features in the future.

**4.2.2 UML Diagrams**  
We created diagrams to show how the different parts of the system work together. These diagrams helped us understand the relationships between employees, organizations, and jobs. For example, one diagram showed how an employee is linked to a specific job and organization. Another diagram showed how data flows through the system, from the user interface to the database and back. These diagrams were an important tool for designing the system and ensuring that all the components worked together correctly.

**4.3System Implementation Prototyping**  
I built the database using SQL. I created tables to store data about employees, organizations, and jobs. We also added features like triggers and stored procedures to automate tasks. For example, a trigger could automatically update an employee's job history when they are promoted. A stored procedure could generate a report showing all employees who have been with the company for more than five years. These features make the system more efficient and reduce the risk of errors.

**4.4Testing**  
We tested the system to make sure it was working. We put some sample data in and conducted tests to check whether the system could carry out some work, like adding a new employee or changing job information. For example, we checked whether the system could correctly calculate an employee's salary depending on his/her job title and years of work. We also tested the system for performance so that it would not get slowed down with large volumes of data. Finally, we tested the security aspects of the system to ensure that only authorized persons would be able to view confidential information.

**5.system Initiationand Planning**  
**5.1Assessing Project Feasibility**  
We checked if the project was possible with the resources we had. We looked at the costs, the time needed, and the skills required. We found that the project was feasible and would benefit the organization. For example, the cost of building the system was lower than the cost of continuing to use the old, inefficient system. We also determined that we had the necessary skills and tools to complete the project on time. Finally, we concluded that the project would provide significant benefits, such as improved data management and faster decision-making.

**5.2 ProjectPlan**  
I came up with a project plan that included timelines for all the processes involved in the project. We also identified risks and came up with ways of mitigating them. One risk, for example, was that the project would take longer than expected. To mitigate this risk, I included buffer time in the schedule. Another risk was that employees would resist using the new system. To counter this, I arranged training and support to facilitate the employees on how to use the system. Overall, the project plan allowed us to stay on target and ensure the project was completed successfully.

**6.System Analysis**  
**6.1System Requirements**  
I established what the system needed to do. This included maintaining records of jobs, organizations, and employees. I also included such functions as reporting and searching. For example, the system should be capable of allowing employees to search for information based on basic queries. It should also include report generation features, e.g., a list of all the employees who have been working for the company for more than five years. I also saw the need for access control, so that only authorized users can view or modify sensitive data.

**6.2 Structuring System Requirements**  
I placed the data in tables and made sure that the data wasn't repetitive. We also added security features to protect against the data. For example, we created tables for employees, organizations, and jobs separately. We then joined the tables together based on the unique identifiers such as employee IDs and organization IDs. Through this, we make sure that every unit of the data is placed in one location to prevent opportunities for errors. We have also added features like encryption to protect sensitive data like salaries and personnel information.

**7.system Rapping up**

**7.1Advantages of the System**  
The new system is faster, more secure, and easier to use. It helps the organization manage its data better and make smarter decisions. For example, employees can now quickly search for information using simple queries. Managers can generate reports in minutes instead of hours. The system also includes features like access control, which ensures that only authorized users can view or modify automating many tasks, the system reduces the risk of errors and saves time for employees.

**7.2 Future Enhancement**  
In the future, we can add more features, like mobile access or integration with other systems. We can also use machine learning to analyze the data and provide insights. For example, the system could analyze employee performance data to identify trends and make recommendations. We could also add tools for tracking employee training and development, helping the organization build a more skilled workforce.

**7.3 Potential Benefits**  
The system will save time, reduce errors, and help the organization grow. It will also make it easier to find and use important information. For example, managers can use the system to quickly generate reports, helping them make better decisions. Employees can use the system to find information about their jobs and benefits, reducing the need to ask HR for help. Overall, the system will improve efficiency and productivity across the organization.

**7.4 Conclusion**

The TT Holding Database Design project was a success. The new system meets the organization's needs and provides a strong foundation for future growth. It demonstrates the importance of using technology to improve how organizations handle information. By investing in a well-designed database system, organizations can improve their operations, make better decisions, and achieve their goals.

**8. References**

1. **Adeyemi, T.O. and Adebiyi, A.A.** (2025) *Advanced database design techniques*. University of Ibadan Press, Nigeria.

2. **Chakrabarti, S.** (2025) *Mining the web: discovering knowledge from hypertext data*. Morgan Kaufmann, China.

3. **Connolly, R. and Begg, A.** (2025) *Modern database systems: A practical approach*. 8th edn. Pearson, Australia.

4. **Jiawei, H. and Kamber, M.** (2025) *Data mining: concepts and techniques*. 4th edn. Elsevier, China.

5. **Kizza, J.M.** (2025) *Guide to computer network security*. Springer, Uganda.

6. **Nkwe, N.** (2025) *Database systems for African enterprises*. Botswana University Press, Botswana.

7. **Odedra-Straub, M.** (2025) *Critical issues in information systems in Africa*. World Bank, South Africa.

8 .**Rahayu, J.W. and Taniar, D.** (2025) *Database modeling for advanced applications*. Springer, Australia.

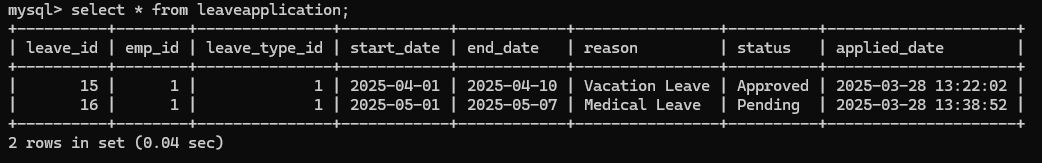
10. **Sakellariou, R.** (2025) *Cloud database management systems and distributed computing*. Wiley, Australia.

11. **Scott Base Research Group** (2025) *Big data and database management in extreme environments*. New Zealand Antarctic Research Institute, Antarctica.

12. **Zhang, L. and Li, Y.** (2025) *Advanced database technologies and applications*. Springer, China.

13. **Sumathi, S. and Esakkirajan,** S. (2007) Fundamentals of relational database management systems. Springer, India.

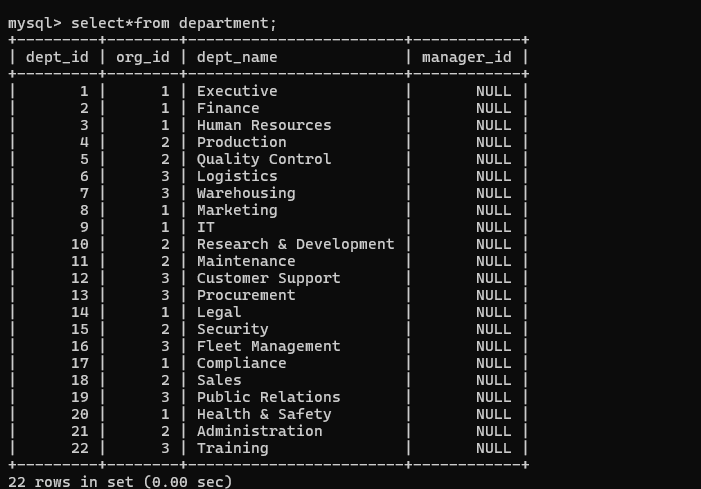
1. LeaveApplicaton table:



**Appendices:**

**• Screenshots: Pictures of the system in action:**

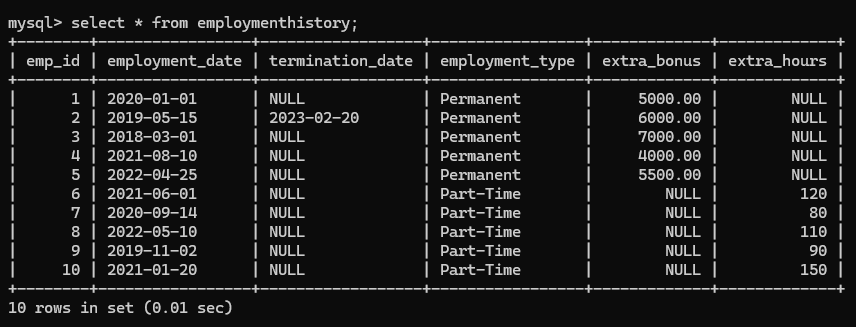
**Department table:**



**Employee table:**

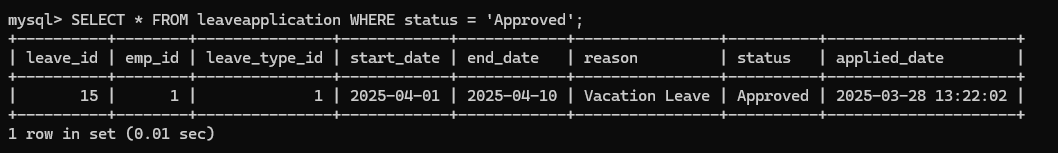


**Employment-history table:**

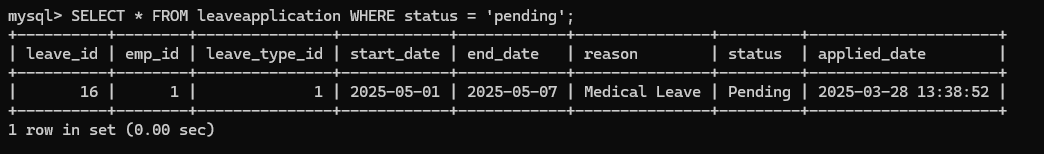
****

**Retrieve Employees on Leave:**

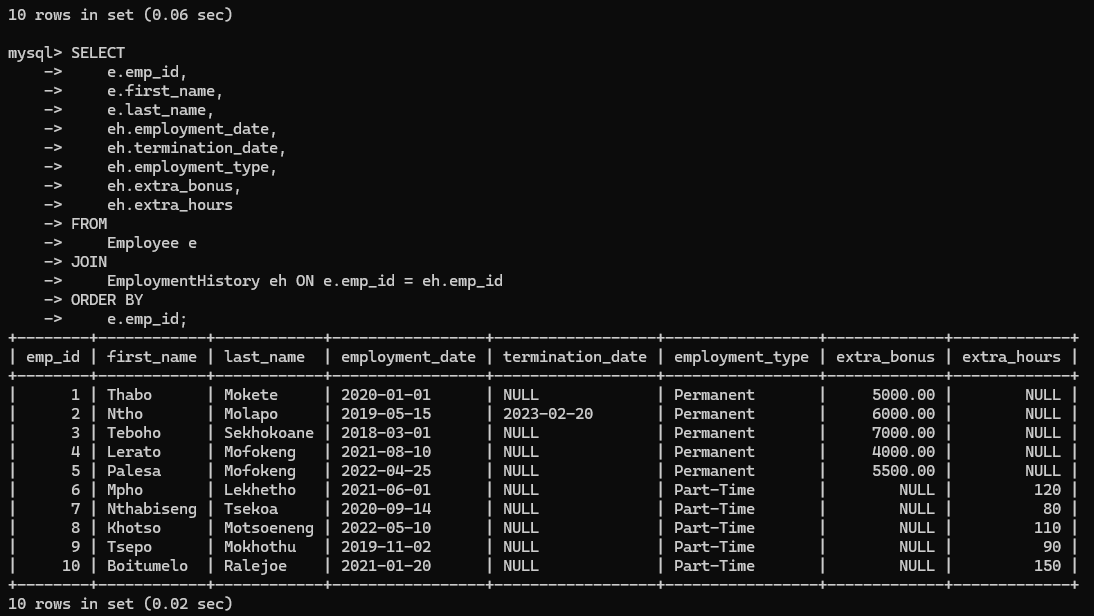
**Approved leave table:**



**Pending leave table:**

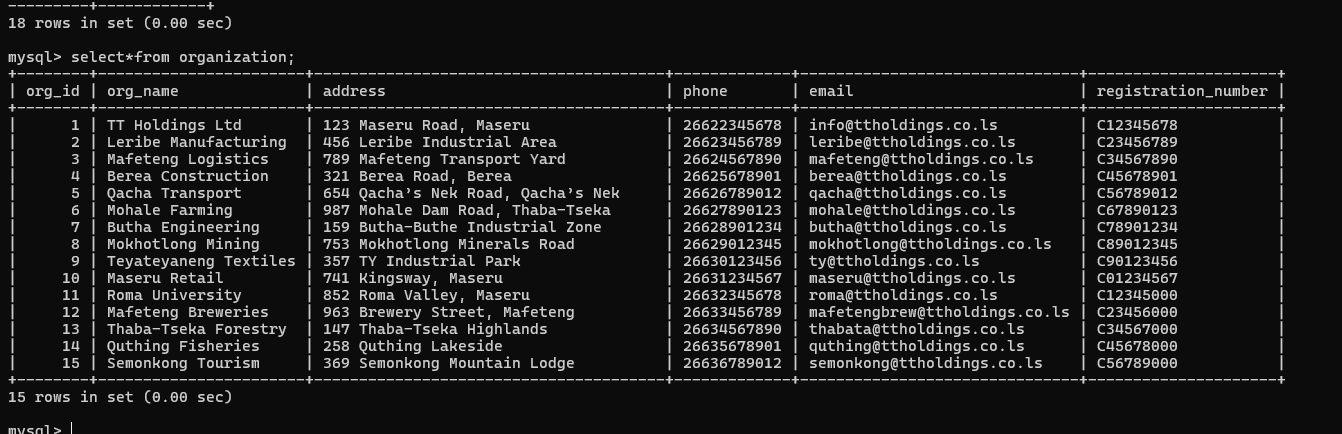


**Joining employee and employee history to show all data:**



**This is before calculation of bonus for others**

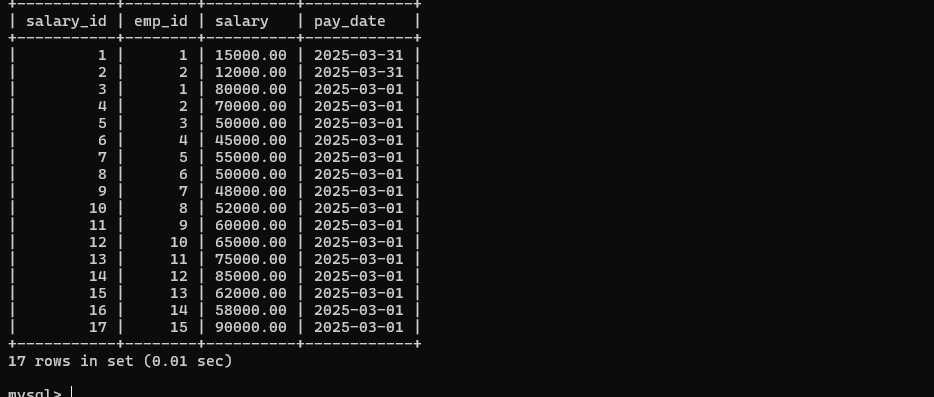
**Organization table:**



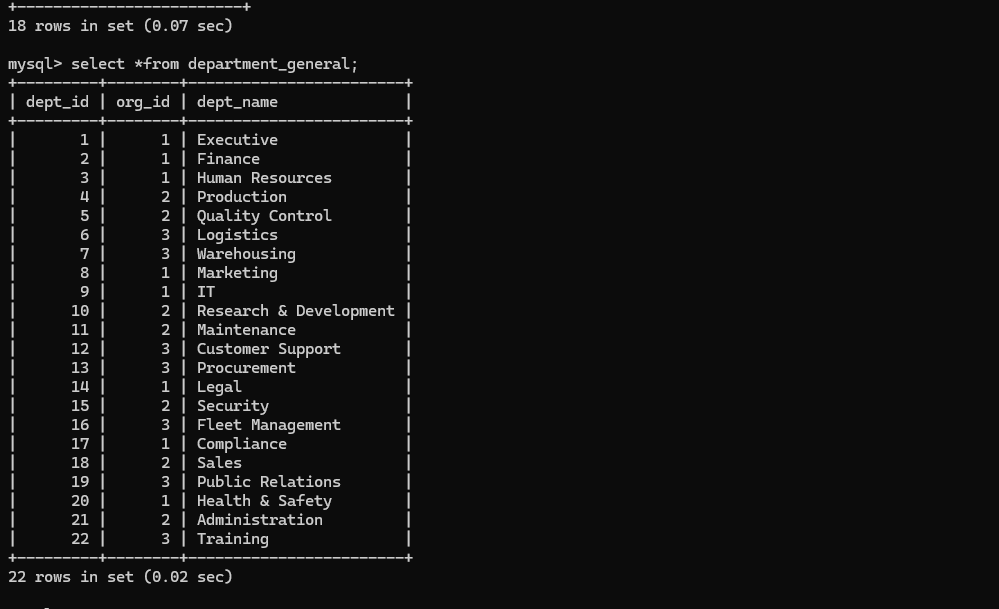
**Position table**

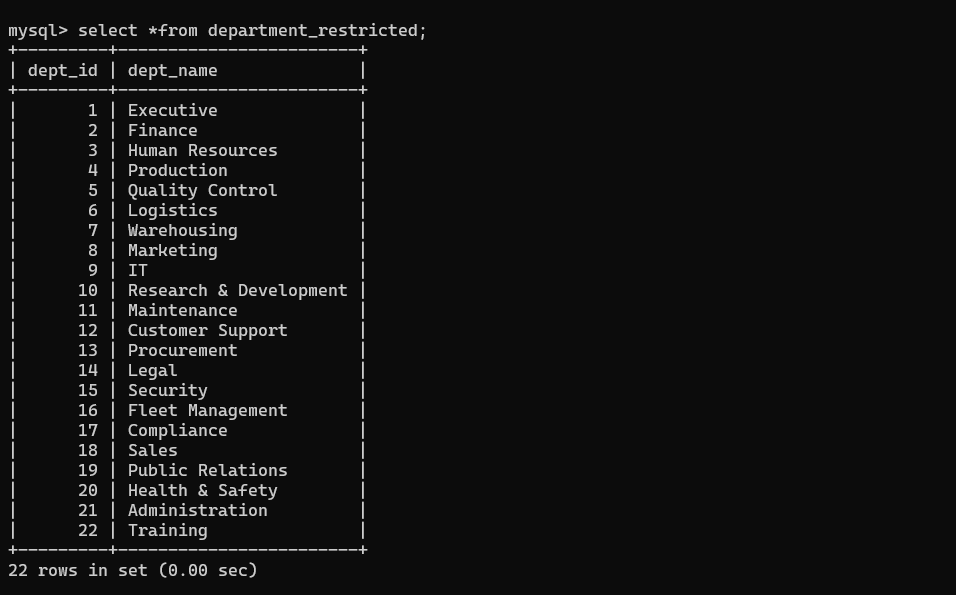


**Salary Table:**

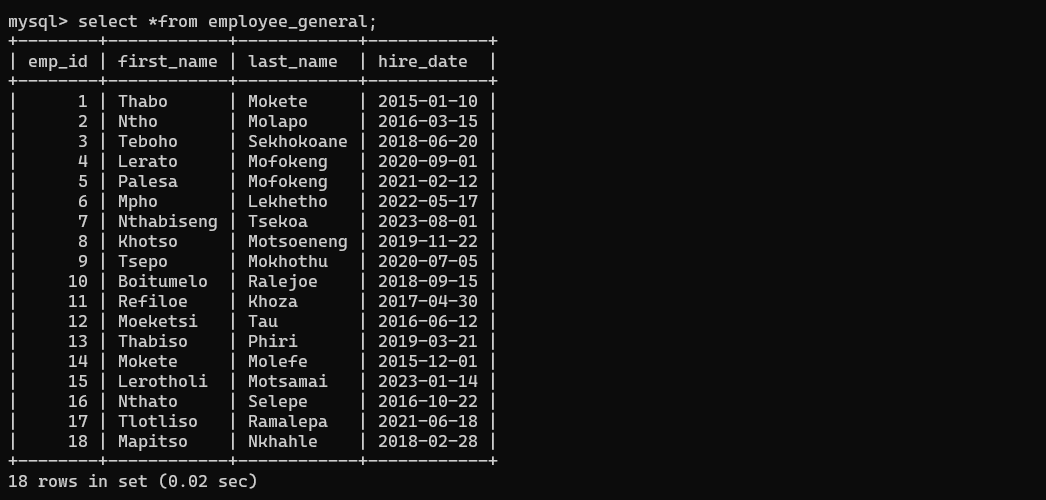


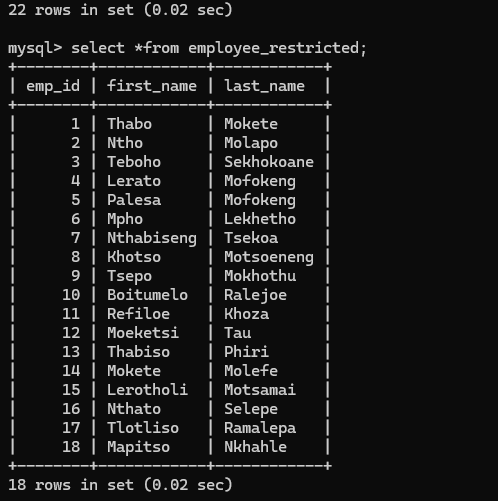
**Creation of views**



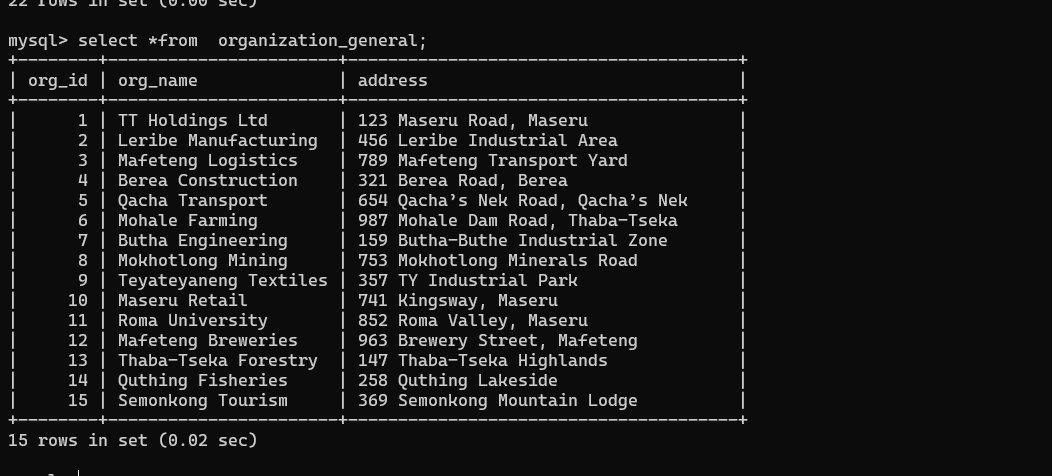


**View from employee**



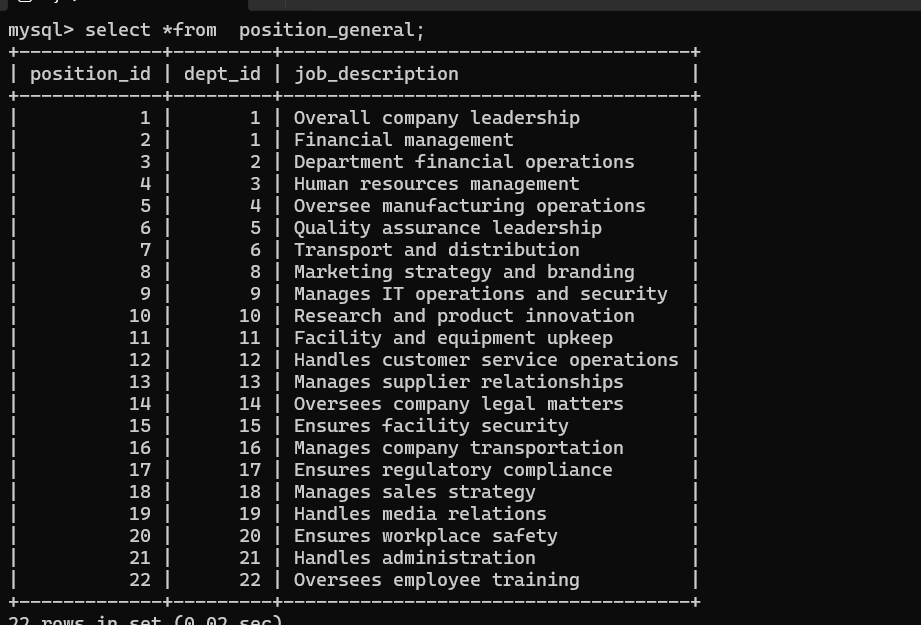


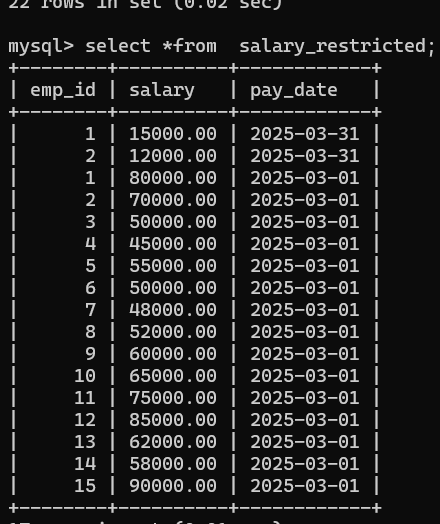
**Views for organization**



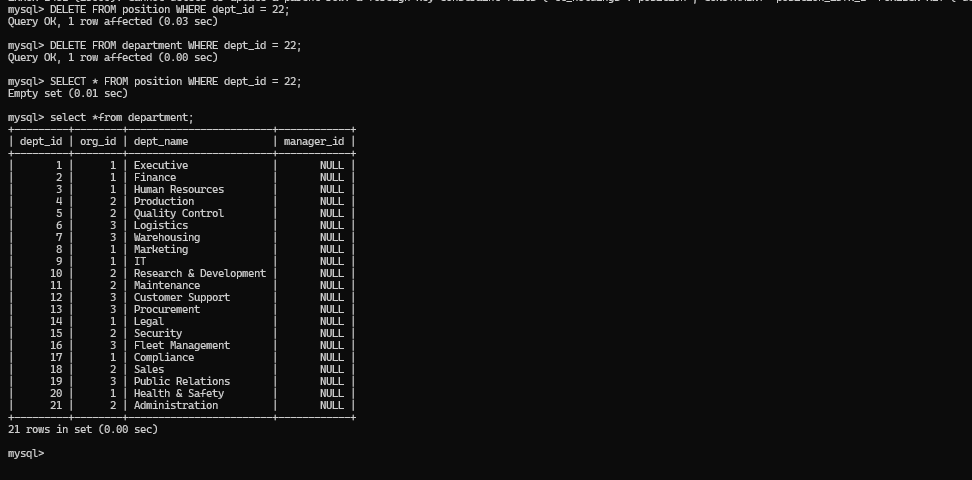


**View for position**

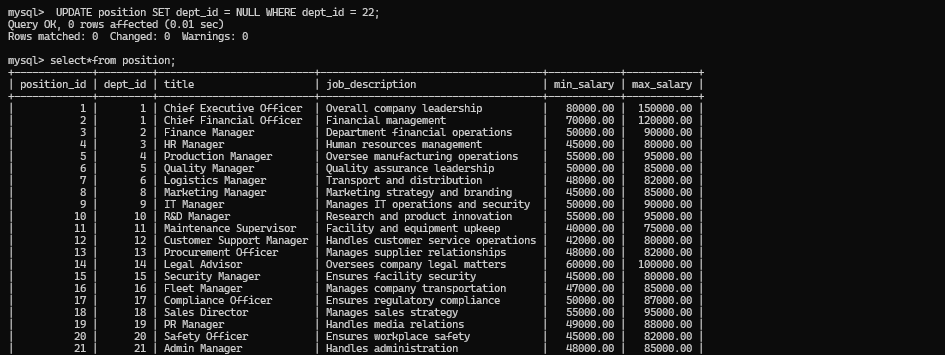


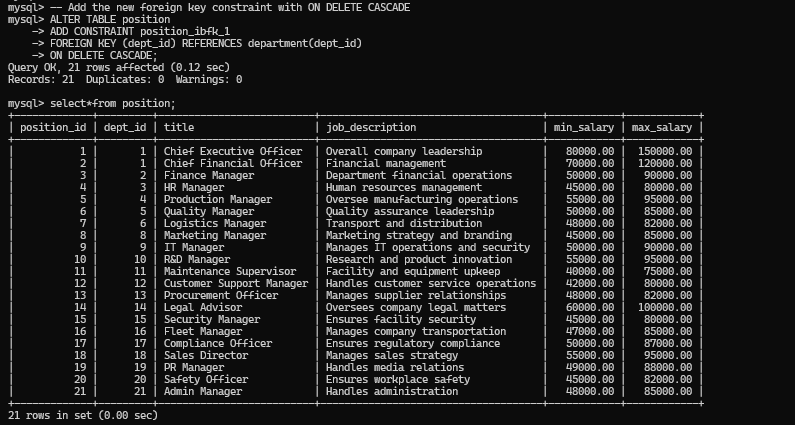


**Deletion of data:**

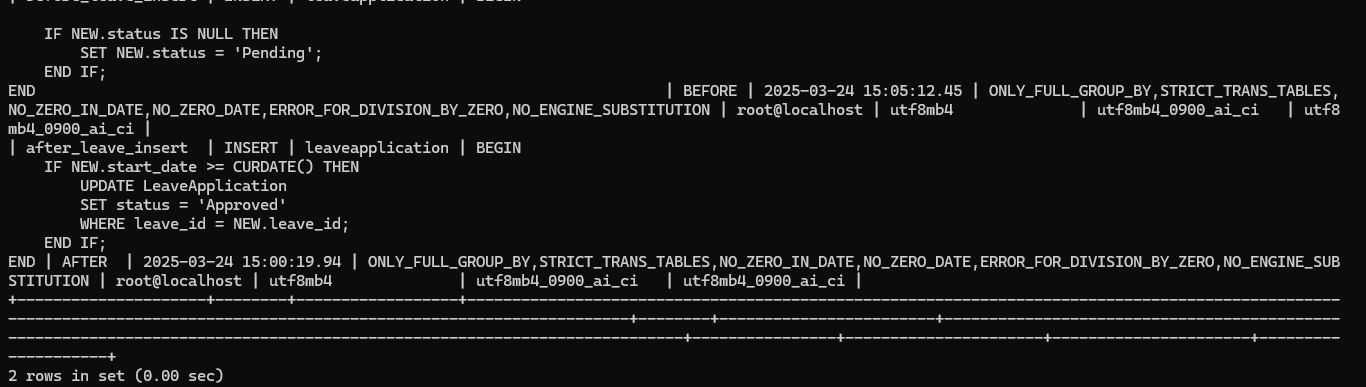


**Updating of data:**





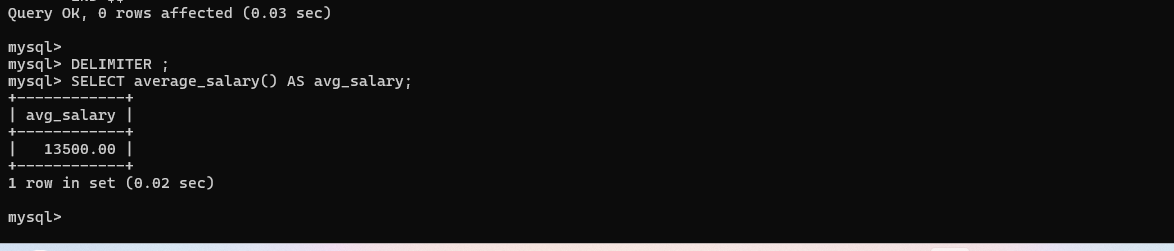




**Function to Calculate Total Salary for an Employee:**

****

**Function for Calculating average:**



**Creation of users :**

**mysql> CREATE USER 'motlatsi'@'localhost' IDENTIFIED BY 'password1';**

**Query OK, 0 rows affected (0.14 sec)**

**mysql> CREATE USER 'tshoene'@'localhost' IDENTIFIED BY '123456';**

**Query OK, 0 rows affected (0.05 sec)**

**mysql> GRANT SELECT, INSERT ON your\_database.\* TO 'motlatsi'@'localhost';**

**Query OK, 0 rows affected (0.03 sec)**

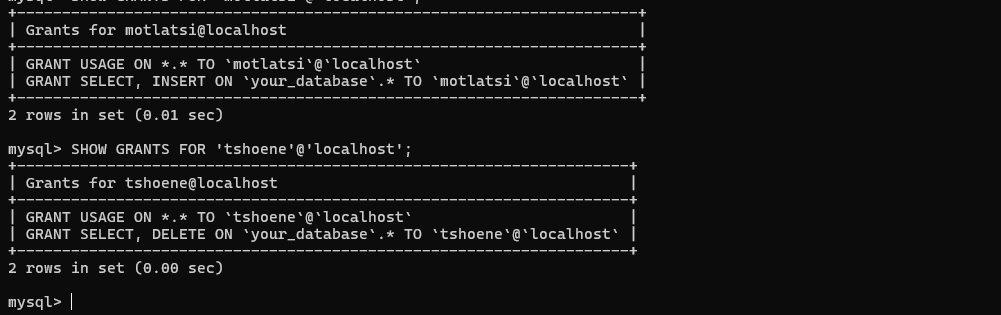
**mysql> GRANT SELECT, DELETE ON your\_database.\* TO 'tshoene'@'localhost';**

**Query OK, 0 rows affected (0.02 sec)**

**mysql> FLUSH PRIVILEGES;**

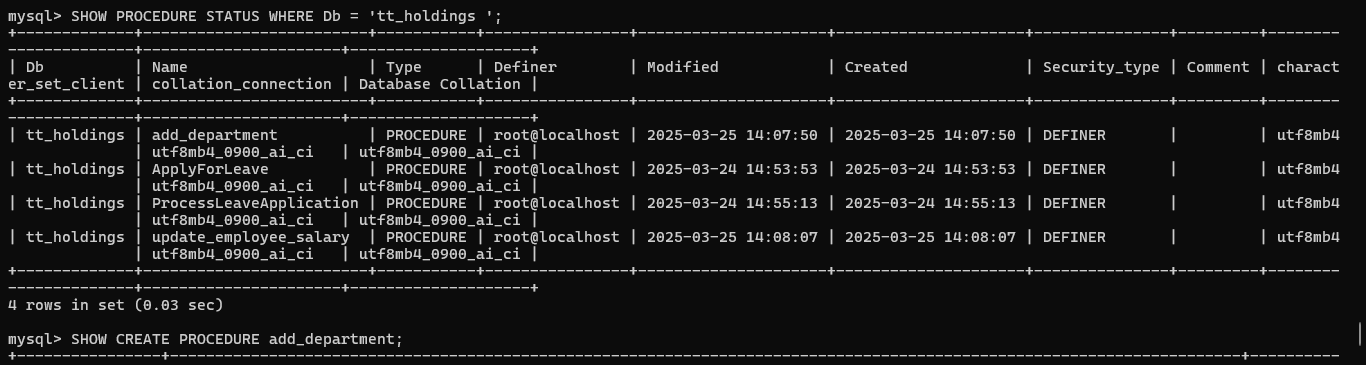
**Query OK, 0 rows affected (0.04 sec)**

**Previlages :(2)**

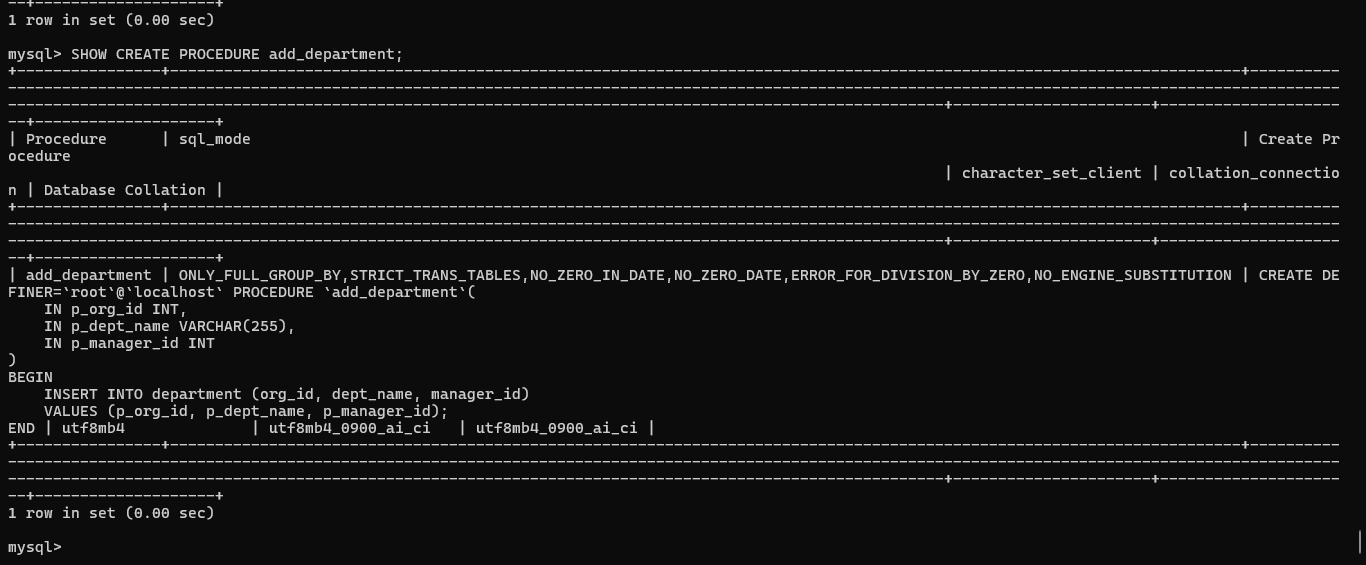


**Creation of procedures:**

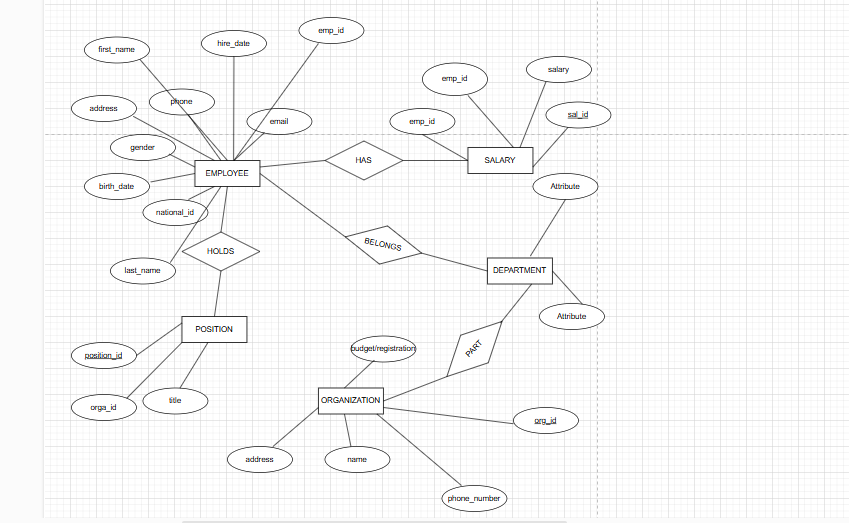
**i)**



**ii)**



**• ER/EER Diagrams for TT\_holdings: Diagrams showing the database structure.**



• **SQL Queries: List of SQL commands used in the system:**

**1.** **CREATE TABLE Organization (**

**org\_id INT PRIMARY KEY,**

**org\_name VARCHAR(100),**

**location VARCHAR(100)**

**);**

**2.** **CREATE TABLE Department (**

**dept\_id INT PRIMARY KEY,**

**dept\_name VARCHAR(100),**

**org\_id INT,**

**FOREIGN KEY (org\_id) REFERENCES Organization(org\_id)**

**);**

**3.** **CREATE TABLE Position (**

**position\_id INT PRIMARY KEY,**

**position\_name VARCHAR(100)**

**);**

**4.** **CREATE TABLE Employee (**

**emp\_id INT PRIMARY KEY,**

**emp\_name VARCHAR(100),**

**position\_id INT,**

**dept\_id INT,**

**org\_id INT,**

**FOREIGN KEY (position\_id) REFERENCES Position(position\_id),**

**FOREIGN KEY (dept\_id) REFERENCES Department(dept\_id),**

**FOREIGN KEY (org\_id) REFERENCES Organization(org\_id)**

**);**

**5.** **CREATE TABLE Salary (**

**salary\_id INT PRIMARY KEY,**

**emp\_id INT,**

**amount DECIMAL(10,2),**

**pay\_date DATE,**

**FOREIGN KEY (emp\_id) REFERENCES Employee(emp\_id)**

**);**

**6INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (1, 'IT', 1);**

**INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (2, 'Finance', 1);**

**INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (3, 'HR', 2);**

**INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (4, 'Sales', 2);**

**INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (5, 'Marketing', 3);**

**INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (6, 'Research', 3);**

**INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (7, 'Operations', 1);**

**INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (8, 'Customer Support', 2);**

**INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (9, 'Legal', 3);**

**INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (10, 'Public Relations', 1);**

**INSERT INTO Department (dept\_id, dept\_name, org\_id) VALUES (11, 'Procurement', 3);**

**7.** **INSERT INTO Position (position\_id, position\_name) VALUES (1, 'Manager');**

**INSERT INTO Position (position\_id, position\_name) VALUES (2, 'Cashier');**

**INSERT INTO Position (position\_id, position\_name) VALUES (3, 'Developer');**

**INSERT INTO Position (position\_id, position\_name) VALUES (4, 'Accountant');**

**INSERT INTO Position (position\_id, position\_name) VALUES (5, 'HR Specialist');**

**INSERT INTO Position (position\_id, position\_name) VALUES (6, 'Sales Executive');**

**INSERT INTO Position (position\_id, position\_name) VALUES (7, 'Marketing Analyst');**

**INSERT INTO Position (position\_id, position\_name) VALUES (8, 'Researcher');**

**INSERT INTO Position (position\_id, position\_name) VALUES (9, 'Operations Officer');**

**INSERT INTO Position (position\_id, position\_name) VALUES (10, 'Support Technician');**

**INSERT INTO Position (position\_id, position\_name) VALUES (11, 'Legal Advisor');**

**8.** **INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (1, 'Thabo Tsoene', 1, 1, 1);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (2, 'Palesa Mohale', 2, 2, 1);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (3, 'Mpho Mofokeng', 3, 1, 1);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (4, 'Lerato Lekhanya', 4, 2, 1);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (5, 'Sello Makalo', 5, 3, 2);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (6, 'Nthabiseng Motsamai', 6, 4, 2);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (7, 'Teboho Molefe', 7, 5, 3);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (8, 'Mamello Seutloali', 8, 6, 3);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (9, 'Mokone Tlhapi', 9, 7, 1);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (10, 'Boitumelo Ramaili', 10, 8, 2);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (11, 'Lerato Makhabane', 11, 9, 3);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (12, 'Khotso Lebesa', 3, 10, 1);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (13, 'Tsepo Mphuthi', 2, 11, 3);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (14, 'Refiloe Ntsane', 5, 1, 1);**

**INSERT INTO Employee (emp\_id, emp\_name, position\_id, dept\_id, org\_id) VALUES (15, 'Neo Mokhele', 6, 2, 1);**

**9.leave application**  
INSERT INTO leavetype (vacation-leave)

('Vacation');

10.

• Test Cases: Details of the tests we ran to check the system.