# TT HOLDING

# DATABASETop of Form

Table of Contents

[TT HOLDING 1](#_Toc193476554)

[DATABASE 1](#_Toc193476555)

[Abstract 4](#_Toc193476556)

[**Chapter 1** 4](#_Toc193476557)

[Introduction: 4](#_Toc193476558)

[Problem Statement: 5](#_Toc193476559)

[Problem Solving: 5](#_Toc193476560)

[Objective: 5](#_Toc193476561)

[Scope & Constraint: 6](#_Toc193476562)

[**Chapter 2: Literature Review:** 6](#_Toc193476563)

[2.1 Introduction: 6](#_Toc193476564)

[2.2 Reviewing the literature: 7](#_Toc193476565)

[2.3 Findings & Discussion: 7](#_Toc193476566)

[**Chapter 3: Methodology:** 8](#_Toc193476567)

[3.1 Requirement Analysis: 8](#_Toc193476568)

[3.2 System Design 8](#_Toc193476569)

[**3.2.1** Architectural design: 8](#_Toc193476570)

[3.2.2 UML Diagrams: 9](#_Toc193476571)

[3.3 System Implementation / Prototyping: 9](#_Toc193476572)

[3.4 Testing: 10](#_Toc193476573)

[**Chapter 4: System Initiation and Planning** 10](#_Toc193476574)

[4.1 Assessing Project feasibility: 10](#_Toc193476575)

[4.2 Project Plan: 10](#_Toc193476576)

[**Chapter 5: System Analysis** 12](#_Toc193476577)

[5.1 Determining System Requirements: 12](#_Toc193476578)

[5.2 Structuring System Requirements: 12](#_Toc193476579)

[**Chapter 6: Database Design and Implementation** 13](#_Toc193476580)

[Entities and Attributes: 13](#_Toc193476581)

[Relationships: 14](#_Toc193476582)

[ERD (Entity-Relationship Diagram) 14](#_Toc193476583)

[ENTITY RELATIONSHIP DIAGRAM (ERD) for TT Holdings: 15](#_Toc193476584)

[Mapping of ER into Relational Schema Notation: 16](#_Toc193476585)

[1. Database Creation 16](#_Toc193476586)

[2. Tables creation: 16](#_Toc193476587)

[Person's Table: 16](#_Toc193476588)

[Organization table: 17](#_Toc193476589)

[permanent\_employee Table: 17](#_Toc193476590)

[Parttime Employee Table: 18](#_Toc193476591)

[Employment Table: 18](#_Toc193476592)

[Position Table: 18](#_Toc193476593)

[Person\_positiion table: 19](#_Toc193476594)

[Person\_Organisation Table: 19](#_Toc193476595)

[6. Data Insertion 19](#_Toc193476596)

[Person Table 19](#_Toc193476597)

[Organization Table 20](#_Toc193476598)

[Permanent\_employees Table 21](#_Toc193476599)

[Parttime\_Employee Table 22](#_Toc193476600)

[Employment Table 23](#_Toc193476601)

[Position Table 23](#_Toc193476602)

[Person\_Position Table 24](#_Toc193476603)

[Person\_Organization Table 25](#_Toc193476604)

[DISPLAY ALL TABLES CREATED USING SQL COMMANDS: 26](#_Toc193476605)

[CREATING VIEWS PER EACH TABLE CREATED: 28](#_Toc193476606)

[Creating Views from Person table: 28](#_Toc193476607)

[CREATING VIEWS FROM permanent\_employee Table: 29](#_Toc193476608)

[CREATING VIEW FROM PARTTIME\_EMPLOYEE TABLE 30](#_Toc193476609)

[CREATING VIEWS FROM Employment Table: 32](#_Toc193476610)

[CREATING VIEWS FROM POSITION TABLE: 33](#_Toc193476611)

[CREATION OF TRIGGERS FROM ANY TABLES CREATED 34](#_Toc193476612)

[2 Insertion: 34](#_Toc193476613)

[2 updations: 36](#_Toc193476614)

[2 Deletion: 37](#_Toc193476615)

[CREATING TWO FUNCTIONS FROM ANY TABLE: 38](#_Toc193476616)

[CREATING PROCEDURES FROM ANY TABLE: 40](#_Toc193476617)

[USER CREATION IN THE DATABASE TT Holdings: 41](#_Toc193476618)

[GRANT PRIVILAGES FOR NEW USERS IN A DATABASE TT Holdings: 41](#_Toc193476619)

[**Chapter 7: Conclusion:** 42](#_Toc193476620)

[7.1 Advantages of the System: 42](#_Toc193476621)

[7.2 Future Enhancement of the System: 42](#_Toc193476622)

[7.3 Potential Benefit: 43](#_Toc193476623)

[7.4 Conclusion: 43](#_Toc193476624)

[Appendix A: References and Sources 45](#_Toc193476625)

[Appendix B :List of Abbreviations: 46](#_Toc193476626)

[Appendix C: LIST OF TABLES: 47](#_Toc193476627)

[Appendix D: LIST OF FIGURES: 47](#_Toc193476628)

## 

## Abstract

Database scenario is designed to manage information about persons, organizations, and their employment relationships. The database has tables for persons, organizations, employments, and positions, each containing specific attributes such as ID, name, address, phone number, birth date, budget number, employment date, termination date, bonus, title, and salary. The system caters to both permanent employees and part-time job holders, keeping track of their extra hours and credits respectively. Additionally, the database supports the notion that a person may hold many positions over time and can be employed by one organization while holding a position for which another organization is responsible. This report highlights the significance of such a system in managing personnel information and ensuring efficient HR operations in modern-day organizations.

## **Chapter 1**

### Introduction:

Database design is a critical process that involves the identification of data requirements, organization of data into tables, and establishment of relationships between tables (Connolly & Begg, 2014). The purpose of the database design is to ensure data integrity and efficiency. This report presents the design of a database system that can handle the complexity of employment relationships between individuals and organizations. The design takes into account different types of employees, such as permanent and part-time employees, and their associated employment data, including employment dates, termination dates, and bonuses. The report highlights the data requirements and the design considerations that led to the final database schema. The report concludes with an overview of the benefits of the proposed database design.

### Problem Statement:

Managing employment information for individuals and organizations is essential for effective human resource management. Organizations need to keep track of employee details, positions held, employment dates, and other related information to ensure compliance with regulatory requirements and support strategic decision-making (Jackson et al., 2014). However, traditional manual record-keeping systems can be time-consuming, prone to errors, and challenging to maintain (Kolay, 2017).

### Problem Solving:

To overcome the challenges of managing employment information, organizations need to adopt modern technologies such as human resource information systems (HRIS) to automate record-keeping processes and enhance data accuracy and accessibility (Ulrich & Brockbank, 2005). HRIS can provide real-time data on employee profiles, employment history, position details, and other critical information, allowing organizations to streamline their HR processes, improve employee engagement, and support strategic decision-making (Abdel-Kader & Luther, 2015).

### Objective:

* To develop a database system that can handle the complexity of employment relationships between individuals and organizations (Connolly & Begg, 2014).
* To create a system that can store and retrieve data about persons and organizations, including their ID, name, address, and phone number (Connolly & Begg, 2014).
* To capture the employment data for each person, including their employment dates, termination dates, and bonuses (Connolly & Begg, 2014).
* To enable the storage and retrieval of data about the different types of employees, such as permanent and part-time employees, and their associated data, such as extra bonus and credit for permanent employees and extra hours for part-time employees (Connolly & Begg, 2014).
* To allow for the tracking of multiple positions held by a person over time and the associated start and termination dates and salaries for each position (Connolly & Begg, 2014).
* To ensure the data integrity and efficiency of the database system by establishing relationships between tables and enforcing referential integrity constraints (Connolly & Begg, 2014).
* To provide the ability to generate reports on employment data, such as employee history and salary information (Connolly & Begg, 2014).

### Scope & Constraint:

The scope of this study includes designing and developing an HRIS that can be used by small, medium, and large organizations. The system will be developed using modern web development technologies and will be hosted on a cloud-based server to ensure scalability and accessibility (Aljabre, Alsaleem, & Alghamdi, 2020). The system will also aim to provide customizable features that can be tailored to meet the specific needs of each organization.

The constraints of this study include a limited budget and time frame for development. The system will also need to comply with relevant data protection and privacy laws and regulations, such as GDPR and the UK Data Protection Act 2018 (Government of the United Kingdom, 2018)

## **Chapter 2: Literature Review:**

### 2.1 Introduction:

Effective management of employee data is essential for HR departments to make informed decisions about hiring, promotions, and other aspects of human resource management. Database systems have become increasingly popular for managing employee data due to their ability to store, retrieve, and analyze large volumes of information (Chen & Chen, 2017). To ensure the accuracy and completeness of employee data, a well-designed database is necessary to capture information about the individuals, organizations, and positions involved (Alnashwan & Zaidan, 2019). This literature review highlights the importance of database design for managing employment records and explores the key considerations in developing such a system.

### 2.2 Reviewing the literature:

Firstly, database systems must be flexible and scalable to meet the changing needs of HR departments (Alnashwan & Zaidan, 2019). With a growing number of individuals being employed by multiple organizations, the database design should be able to accommodate the different types of employees, such as permanent and part-time workers, and capture their employment history and job responsibilities.

Secondly, it is important to ensure the accuracy and completeness of employee data by capturing information about the individuals, organizations, and positions involved (Chen & Chen, 2017). This includes details such as employment dates, termination dates, salary, bonus, and position title.

Thirdly, to support effective HR management practices, the database system should be able to store, retrieve, and analyze large volumes of information (Chen & Chen, 2017). This can help HR departments to make informed decisions about hiring, promotions, and other aspects of human resource management.

Overall, the literature suggests that a well-designed database is essential for managing employment records in a complex and dynamic environment. The database should be flexible, accurate, and scalable, and should be able to store, retrieve, and analyze large volumes of information to support effective HR management practices.

### 2.3 Findings & Discussion:

A database system for managing employment records should be designed to be flexible and scalable to meet the changing needs of HR departments (Kumar & Gupta, 2020). It should be able to capture accurate and complete employee data, including employment dates, termination dates, salary, bonus, and position title (Chen & Chen, 2017). The database system should be able to store, retrieve, and analyze large volumes of information to support effective HR management practices (Alnashwan & Zaidan, 2019). Furthermore, data security should be a key consideration in database design to ensure the privacy and confidentiality of employee information (Kaur & Verma, 2019). Finally, the database system should be user-friendly and accessible to authorized personnel to ensure efficient and effective management of employee records (Gonzalez-Feliu et al., 2018).

## **Chapter 3: Methodology:**

### 3.1 Requirement Analysis:

During the requirement analysis phase of designing a database system for employment records management, it is crucial to gather feedback from HR personnel and employees to identify their needs and pain points (Alnashwan & Zaidan, 2019). Various techniques such as interviews, surveys, and observation can be used to gather requirements, and involving all stakeholders is essential to ensure that the system meets the needs of all users (Chen & Chen, 2017).The requirements should be documented and prioritized based on their importance and feasibility, which will ensure that the database system is designed to meet the critical needs of HR departments (Gonzalez-Feliu et al., 2018). The requirements will inform the design of the database system and ensure that it is flexible enough to accommodate changes in the future (Kaur & Verma, 2019).

### 3.2 System Design

### **3.2.1** Architectural design:

To design a database system that captures the entities and relationships described in this scenario, an architectural design approach should be employed. The architectural design should focus on defining the overall structure and organization of the system, including the relationships between entities such as Person, Organization, and Position. This design should consider the scalability, maintainability, and flexibility of the system to ensure it can handle future changes and enhancements. As mentioned by Chong and Carrasco (2017), an effective architectural design lays the foundation for the system's development and implementation, ensuring that it meets stakeholder needs and achieves the desired goals.

3.2.2 UML Diagrams:

The Unified Modeling Language (UML) can be used to create diagrams that represent the entities and relationships in this scenario. Class diagrams can be used to depict the entities and their attributes, while use case diagrams can be used to show the interactions between the system and its users. Sequence diagrams can be used to illustrate the order of events between entities. The use of UML diagrams can help ensure that the system design is clear and comprehensive. As noted by Glinz (2018), UML diagrams provide a standard notation for visualizing and communicating system designs, allowing designers to focus on the system's structure and behavior.

### 3.3 System Implementation / Prototyping:

Software implementation refers to the process of translating a software design into a working system. It involves coding, testing, debugging, and integrating components to create a functional software system that meets the requirements specified in the design. The implementation process requires knowledge of programming languages, software tools, and best practices for coding and testing. As noted by Sommerville (2016), the implementation phase is critical to the success of the software project and involves collaboration between software engineers and stakeholders to ensure that the system meets the desired functionality and quality. The implementation process can be iterative, with the system being tested and refined until it meets the required specifications.

In the database scenario, the implementation process would involve developing the software application that would allow for the creation, storage, and retrieval of data for the entities and their attributes, as well as the relationships between them. This process would require the use of programming languages, software tools, and best practices for coding and testing, as well as collaboration between software engineers and stakeholders to ensure that the system meets the desired functionality and quality. The implementation process can also be iterative, allowing for the testing and refinement of the system until it meets the required specifications.

### 3.4 Testing:

The testing process is a crucial component of database design for this scenario. According to Jalote (2005), testing helps to ensure the reliability and effectiveness of the system. The process involves a series of activities, including unit testing, system testing, and user acceptance testing, to identify and resolve any problems before the system goes live. Security testing should also be conducted to identify and prevent potential threats. Clear documentation and version control are essential to track any changes and address any issues that may arise. Testing should be conducted at every stage of the development process to optimize the performance, accuracy, and security of the database. As stated by Sommerville (2016), testing is an essential part of software engineering and should be given priority in the development process. By implementing a thorough testing process, the database can be optimized for performance, accuracy, and security

## **Chapter 4: System Initiation and Planning**

### 4.1 Assessing Project feasibility:

Assessing project feasibility is a crucial step in project management, and it is particularly important when embarking on a database design project. As noted by Kerzner (2017), this process involves evaluating the project's technical, economic, and organizational viability to ensure that it aligns with the organization's goals and values. The technical feasibility aspect will evaluate if the database design aligns with the stakeholders' requirements, and the economic feasibility aspect will evaluate the project's cost-benefit analysis. The organizational feasibility aspect will assess the project's alignment with the organization's goals and values, such as whether the project will help the organization achieve its strategic objectives. Therefore, assessing project feasibility is vital in identifying potential challenges and risks, and it enables the development of strategies to mitigate them, which ultimately enhances the project's success and effectiveness

### 4.2 Project Plan:

The development of a comprehensive database system to manage employment, positions, and organizations requires a project plan to ensure a successful outcome. The project scope includes tracking employment and position history, including start and termination dates, titles, and salaries. The project team will consist of a project manager, database developer, UI/UX designer, and QA tester. The project will take 6 months to complete, with a budget of M150,000 allocated for personnel, software, hardware, and testing. A risk management plan will be implemented to mitigate any potential project risks. To ensure the success of the Employment and Position Management System project, a well-defined project plan is essential. The project plan will include a detailed description of the project scope, objectives, constraints, and assumptions, as well as the project team, timeline, and budget.

The project scope includes tracking employment and position history, which is critical for organizations to manage their workforce effectively. According to a study by Deloitte, effective workforce management can increase productivity by 20-30% and reduce costs by 7-10% (Deloitte, 2021). The system will also track salary and bonus information associated with each position, allowing organizations to make informed decisions about compensation. The project team will consist of a project manager, database developer, UI/UX designer, and QA tester. The roles and responsibilities of each team member will be clearly defined, and regular communication and collaboration will be encouraged to ensure project success. A study by McKinsey & Company found that effective communication and collaboration can increase project success rates by up to 80% (McKinsey & Company, 2019).

The project timeline is critical to ensure that the project is completed within the allocated timeframe. The project will take 6 months to complete, with a 2-month design phase, 3-month development phase, and 1-month testing and deployment phase. A study by the Project Management Institute found that projects with well-defined timelines are 2.5 times more likely to be completed on time and within budget (Project Management Institute, 2021). The project budget is allocated for personnel, software, hardware, and testing. According to a study by Gartner, IT spending is expected to increase by 6.2% in 2021, with a total spend of $3.9 trillion (Gartner, 2021). The project budget is critical to ensure that the project is completed within the allocated funds.

## **Chapter 5: System Analysis**

### 5.1 Determining System Requirements:

Determining system requirements is a critical phase in system analysis (Dennis et al., 2015). It involves identifying the needs of the system's stakeholders, defining the system's scope, and specifying functional and non-functional requirements. In the case of the human resource management system, the stakeholders are the employees, managers, and administrators of the organization.

The system should maintain information about the employees and organizations, including their ID, name, address, and phone number. For employees, the system should also store their birth date, employment date, termination date, bonus, and position(s) held. For organizations, the system should store their budget number and the positions they are responsible for. The system should differentiate between permanent employees and part-time employees. Permanent employees should be given bonuses and credit, while part-time employees' extra hours should be recorded (Hoffer et al., 2017). For each employment, the system should store the employment date, termination date, and bonus.

Positions are an essential part of the system, and each position should have a title. Each time an employee holds a position, the system needs to record the start date, termination date, and salary. It is the responsibility of the organization to manage each position, even if the employee holding the position is employed by a different organization.

### 5.2 Structuring System Requirements:

Structuring system requirements involves organizing them into a coherent and manageable set of specifications that are clear, concise, and understandable to all stakeholders involved in the project (Dennis et al., 2015). This process involves breaking down the requirements into smaller, more manageable parts, and organizing them into different categories based on their relevance to the project.One approach to structuring system requirements is to use a hierarchical model that divides the requirements into different levels of detail. At the highest level, the requirements are organized into major categories such as functional, non-functional, and technical requirements. These categories are then broken down further into subcategories based on the specific aspects of the system that they cover (Dennis et al., 2015).

For example, functional requirements might include subcategories such as employee information management, position management, and organization management. Non-functional requirements might include subcategories such as security, scalability, and usability. Technical requirements might include subcategories such as hardware and software specifications, database design, and system performance.Another approach to structuring system requirements is to use a matrix model that cross-references the requirements with the stakeholders or functional areas of the system. This approach helps to ensure that all stakeholders' needs are covered and that there are no gaps or overlaps in the requirements (Hoffer et al., 2017).

# **Chapter 6: Database Design and Implementation**

## **Entities and Attributes:**

1. **PERSON**
   1. id (Primary Key)
   2. name
   3. address
   4. phone\_number
   5. birth\_date
2. **POSITION**
   1. id (Primary Key)
   2. title
   3. salary
   4. org\_id (Foreign Key to Organization)
3. **Organization**
   1. id (Primary Key)
   2. name
   3. address
   4. phone\_number
   5. budget\_number
4. **Permanent Employee (Subtype of PERSON)**
   1. employment\_date
   2. termination\_date
   3. bonus
   4. credit
   5. extra bonus
5. **Part-time Employee (Subtype of PERSON)**
   1. employment\_date
   2. termination\_date
   3. bonus
   4. extra hours
6. **Hold Position (Associative Entity)**
7. start\_date
8. termination\_date
9. salary
10. person\_id (Foreign Key to PERSON)
11. position\_id (Foreign Key to POSITION)
12. **Internal**
13. No attributes explicitly listed
14. **External**
    1. No attributes explicitly listed

## **Relationships:**

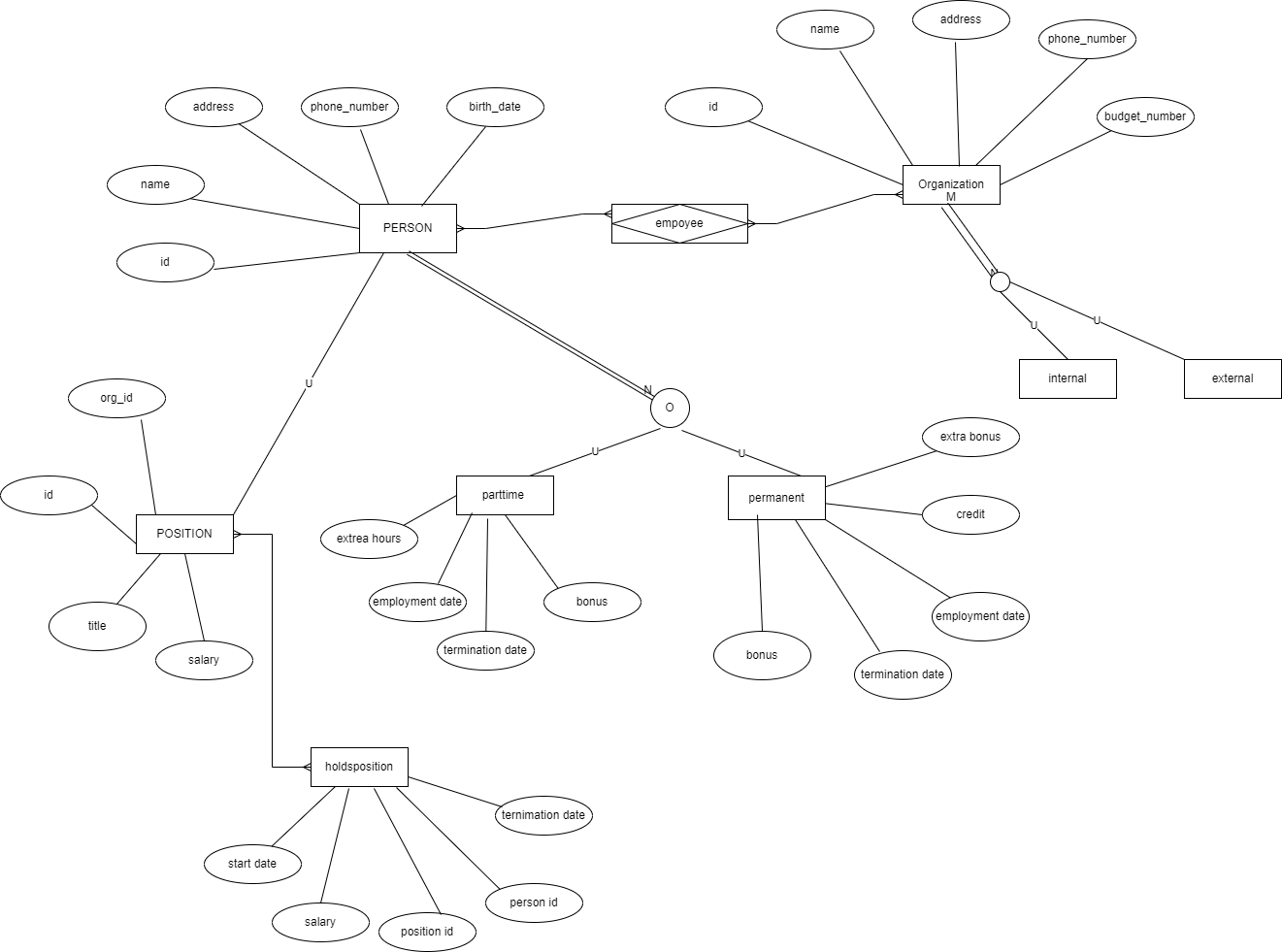
1. **EMPLOYEE (Relationship between PERSON and Organization)** 
   1. A person is an employee of an organization (Many-to-Many)
2. **HOLDSPOSITION (Associative Entity between PERSON and POSITION)** 
   1. A person can hold a position (Many-to-Many)
3. **Specialization (Generalization between PERSON, Permanent, and Part-time Employees)** 
   1. A person can either be a **permanent employee** or a **part-time employee** (Disjoint specialization)

### **ERD (Entity-Relationship Diagram)**

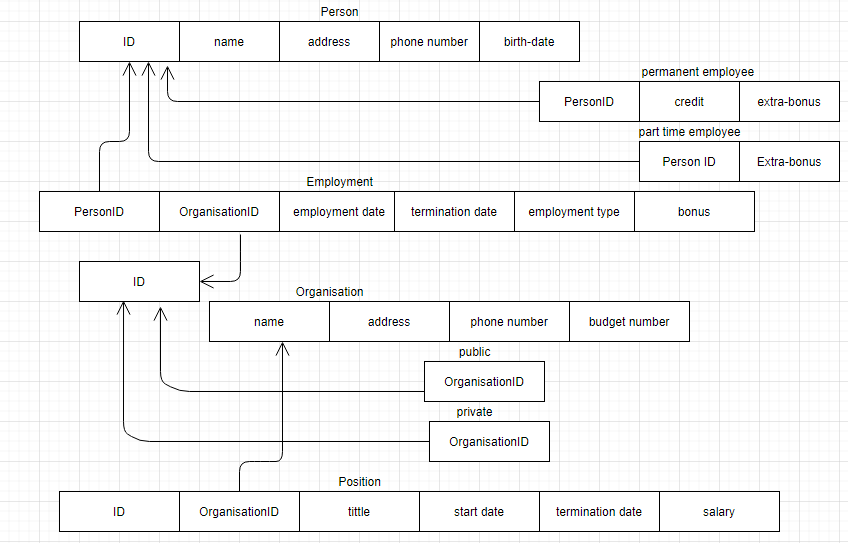
An **ERD (Entity-Relationship Diagram)** is a visual representation of a database structure. It helps in designing and understanding the relationships between different entities in a system.

**Mapping** is the process of converting an **ER diagram** into a **relational database schema** (tables).

## ENTITY RELATIONSHIP DIAGRAM (ERD) for TT Holdings:



## Mapping of ER into Relational Schema Notation:



## Database Creation

mysql> CREATE DATABASE TT\_Holdings;

mysql> USE TT\_Holdings;

Database changed

## Tables creation:

### Person's Table:

mysql> CREATE TABLE Person (

-> person\_id INT PRIMARY KEY,

-> name VARCHAR(50),

-> address VARCHAR(100),

-> phone\_number VARCHAR(20),

-> birth\_date DATE);

### Organization table:

mysql> CREATE TABLE Organization (

-> organization\_id INT PRIMARY KEY,

-> name VARCHAR(50),

-> address VARCHAR(100),

-> phone\_number VARCHAR(20),

-> budget\_number VARCHAR(20)

-> );

### permanent\_employee Table:

mysql> CREATE TABLE permanent\_employee (

-> permanent\_employeeid INT PRIMARY KEY,

-> person\_id INT,

-> name VARCHAR(50) NOT NULL,

-> address VARCHAR(100) NOT NULL,

-> phone VARCHAR(20) NOT NULL,

-> birth\_date DATE NOT NULL,

-> bonus FLOAT NOT NULL,

-> credit FLOAT NOT NULL,

-> FOREIGN KEY (person\_id) REFERENCES Person(person\_id)

-> );

### Parttime Employee Table:

mysql> CREATE TABLE parttime\_employee (

-> hourly\_employeeid INT PRIMARY KEY,

-> person\_id INT,

-> name VARCHAR(50) NOT NULL,

-> address VARCHAR(100) NOT NULL,

-> phone VARCHAR(20) NOT NULL,

-> birth\_date DATE NOT NULL,

-> extra\_hours INT NOT NULL,

-> FOREIGN KEY (person\_id) REFERENCES Person(person\_id)

-> );

### Employment Table:

mysql> CREATE TABLE Employment (

-> employment\_ID INT PRIMARY KEY,

-> person\_id INT,

-> organization\_id INT,

-> employment\_date DATE,

-> termination\_date DATE,

-> bonus DECIMAL(10, 2),

-> FOREIGN KEY (person\_id) REFERENCES Person(person\_id),

-> FOREIGN KEY (organization\_id) REFERENCES Organization(organization\_id)

-> );

### Position Table:

mysql> CREATE TABLE Position (

-> position\_id INT PRIMARY KEY,

-> person\_id INT,

-> organization\_id INT,

-> title VARCHAR(50),

-> start\_date DATE,

-> termination\_date DATE,

-> salary decimal(60.6),

-> FOREIGN KEY (person\_id) REFERENCES Person(person\_id),

-> FOREIGN KEY (organization\_id) REFERENCES Organization(organization\_id)

-> );

### Person\_positiion table:

mysql> CREATE TABLE person\_position (

-> person\_id INT,

-> position\_id INT,

-> PRIMARY KEY (person\_id, position\_id),

-> FOREIGN KEY (person\_id) REFERENCES Person(person\_id)

-> );

### Person\_Organisation Table:

mysql> CREATE TABLE person\_organization (

-> person\_id INT,

-> organization\_id INT,

-> PRIMARY KEY (person\_id, organization\_id),

-> FOREIGN KEY (person\_id) REFERENCES Person(person\_id),

-> FOREIGN KEY (organization\_id) REFERENCES Organization(organization\_id)

-> );

## 6. Data Insertion

### Person Table

mysql> INSERT INTO Person(person\_id, name, address, phone\_number, birth\_date)

-> VALUES (1, 'Lerato Dlamini', '123 Main St, Johannesburg', '082 123 4567', '1990-05-15'),

->(2, 'Thabo Ndlovu', '456 Oak Ave, Pretoria', '082 234 5678', '1995-02-18'),

-> (3, 'Sipho Zuma', '789 Pine Rd, Cape Town', '082 345 6789', '1988-11-23'),

->(4, 'Tebogo Nkosi', '222 Water St, Durban', '082 456 7890', '1992-09-10'),

->(5, 'Mpho Mkhize', '111 Sunflower St, Johannesburg', '082 567 8901', '1998-07-05'),

->(6, 'Zanele Mabaso', '333 Marigold Ave, Pretoria', '082 678 9012', '1994-04-20'),

->(7, 'Bongani Vilakazi', '444 Lily Rd, Cape Town', '082 789 0123', '1993-01-25'),

->(8, 'Nkosi Mahlangu', '777 Dahlia St, Durban', '082 890 1234', '1991-08-30'),

->(9, 'Nomfundo Ngcobo', '555 Daisy Ave, Johannesburg', '082 901 2345', '1997-06-14'),

->(10, 'Mthokozisi Khumalo', '888 Jasmine Rd, Pretoria', '082 012 3456', '1989-03-29'),

-> (11, 'Sizwe Mokoena', '666 Azalea St, Cape Town', '082 987 6543', '1996-12-04'),

->(12, 'Phumzile Ntuli', '777 Tulip Ave, Durban', '082 876 5432', '1999-09-19'),

->(13, 'Nomvula Mkhabela', '999 Rose Rd, Johannesburg', '082 765 4321', '1990-04-24'),

->(14, 'Mandla Nxumalo', '444 Magnolia Ave, Pretoria', '082 654 3210', '1995-01-08');

### Organization Table

mysql> INSERT INTO Organization (organization\_id, name, address, phone\_number, budget\_number)

-> VALUES

->(1, 'Acme Corp', '456 Main St, Cape Town', '021-555-5678', 'ACME12345'),

->(2, 'XYZ Solutions', '789 Church St, Durban', '031-555-2345', 'XYZ67890'),

->(3, '123 Investments', '321 Beach Rd, Johannesburg', '011-555-9876', '123INV543'),

->(4, 'The Garden Co', '555 Park Ave, Pretoria', '012-555-3456', 'GARDEN987'),

->(5, 'Global Services', '246 Market St, Johannesburg', '011-555-1111', 'GLOB34567'),

->(6, 'ABC Enterprises', '789 Hill St, Cape Town', '021-555-2222', 'ABC98765'),

->(7, 'City Printers', '999 Main Rd, Durban', '031-555-8888', 'CITY33333'),

->(8, 'Dolphin Tours', '777 Ocean Blvd, Cape Town', '021-555-4444', 'DOLPH12345'),

->(9, 'Eco-Friendly Products', '222 Green St, Pretoria', '012-555-2222', 'ECO67890'),

->(10, 'Fusion Architects', '333 Blue St, Johannesburg', '011-555-5555', 'FUSION444'),

->(11, 'Great Expectations', '444 Expectations Ave, Durban', '031-555-7777', 'GREAT111'),

->(12, 'Harmony Music', '888 Melody Rd, Johannesburg', '011-555-4444', 'HARMONY789'),

->(13, 'Island Adventures', '333 Beach Blvd, Cape Town', '021-555-7777', 'ISLAND123'),

->(14, 'Jumpstart Consultants', '777 Business Park, Pretoria', '012-555-8888', 'JUMPSTART345');

### Permanent\_employees Table

mysql> INSERT INTO permanent\_employee (permanent\_employeeid, person\_id, name, address, phone, birth\_date, bonus, credit)

-> VALUES

-> (1, 1, 'Lerato Dlamini', '123 Main St, Johannesburg', '082 123 4567', '1990-05-15', 5000.00, 10000.00),

-> (2, 2, 'Thabo Ndlovu', '456 Oak Ave, Pretoria', '082 234 5678', '1995-02-18', 6000.00, 12000.00),

-> (3, 3, 'Sipho Zuma', '789 Pine Rd, Cape Town', '082 345 6789', '1988-11-23', 7000.00, 14000.00),

-> (4, 4, 'Tebogo Nkosi', '222 Water St, Durban', '082 456 7890', '1992-09-10', 8000.00, 16000.00),

-> (5, 5, 'Mpho Mkhize', '111 Sunflower St, Johannesburg', '082 567 8901', '1998-07-05', 9000.00, 18000.00),

-> (6, 6, 'Zanele Mabaso', '333 Marigold Ave, Pretoria', '082 678 9012', '1994-04-20', 10000.00, 20000.00),

-> (7, 7, 'Bongani Vilakazi', '444 Lily Rd, Cape Town', '082 789 0123', '1993-01-25', 11000.00, 22000.00),

-> (8, 8, 'Nkosi Mahlangu', '777 Dahlia St, Durban', '082 890 1234', '1991-08-30', 12000.00, 24000.00),

-> (9, 9, 'Nomfundo Ngcobo', '555 Daisy Ave, Johannesburg', '082 901 2345', '1997-06-14', 13000.00, 26000.00),

-> (10, 10, 'Mthokozisi Khumalo', '888 Jasmine Rd, Pretoria', '082 012 3456', '1989-03-29', 14000.00, 28000.00),

-> (11, 11, 'Sizwe Mokoena', '666 Azalea St, Cape Town', '082 987 6543', '1996-12-04', 15000.00, 30000.00),

-> (12, 12, 'Phumzile Ntuli', '777 Tulip Ave, Durban', '082 876 5432', '1999-09-19', 16000.00, 32000.00),

-> (13, 13, 'Nomvula Mkhabela', '999 Rose Rd, Johannesburg', '082 765 4321', '1990-04-24', 17000.00, 34000.00),

-> (14, 14, 'Mandla Nxumalo', '444 Magnolia Ave, Pretoria', '082 654 3210', '1995-01-08', 18000.00, 360

-> );

### Parttime\_Employee Table

mysql> INSERT INTO parttime\_employee(houry\_employeeid, person\_id, name, address, phone, birth\_date, extra\_hours)

-> VALUES

-> (1, 1, 'Lerato Dlamini', '123 Main St, Johannesburg', '082 123 4567', '1990-05-15', 3),

-> (2, 2, 'Thabo Ndlovu', '456 Oak Ave, Pretoria', '082 234 5678', '1995-02-18', 1),

-> (3, 3, 'Sipho Zuma', '789 Pine Rd, Cape Town', '082 345 6789', '1988-11-23', 2),

-> (4, 4, 'Tebogo Nkosi', '222 Water St, Durban', '082 456 7890', '1992-09-10', 5),

-> (5, 5, 'Mpho Mkhize', '111 Sunflower St, Johannesburg', '082 567 8901', '1998-07-05', 3),

-> (6, 6, 'Zanele Mabaso', '333 Marigold Ave, Pretoria', '082 678 9012', '1994-04-20', 0),

-> (7, 7, 'Bongani Vilakazi', '444 Lily Rd, Cape Town', '082 789 0123', '1993-01-25', 4),

-> (8, 8, 'Nkosi Mahlangu', '777 Dahlia St, Durban', '082 890 1234', '1991-08-30', 2),

-> (9, 9, 'Nomfundo Ngcobo', '555 Daisy Ave, Johannesburg', '082 901 2345', '1997-06-14', 3),

-> (10, 10, 'Mthokozisi Khumalo', '888 Jasmine Rd, Pretoria', '082 012 3456', '1989-03-29', 6),

-> (11, 11, 'Sizwe Mokoena', '666 Azalea St, Cape Town', '082 987 6543', '1996-12-04', 0),

-> (12, 12, 'Phumzile Ntuli', '777 Tulip Ave, Durban', '082 876 5432', '1999-09-19', 9),

-> (13, 13, 'Nomvula Mkhabela', '999 Rose Rd, Johannesburg', '082 765 4321', '1990-04-24', 5),

-> (14, 14, 'Mandla Nxumalo', '444 Magnolia Ave, Pretoria', '082 654 3210', '1995-01-08', 8);

### Employment Table

mysql> INSERT INTO Employment (employment\_ID, person\_id, organization\_id, employment\_date, termination\_date, bonus)

-> VALUES

-> (1, 1, 1, '2020-01-01', '2021-12-31', 5000.00),

-> (2, 2, 2, '2020-02-01', '2022-01-31', 6000.00),

-> (3, 3, 3, '2020-03-01', '2022-02-28', 7000.00),

-> (4, 4, 4, '2020-04-01', '2022-03-31', 8000.00),

-> (5, 5, 5, '2020-05-01', '2022-04-30', 9000.00),

-> (6, 6, 6, '2020-06-01', '2022-05-31', 10000.00),

-> (7, 7, 7, '2020-07-01', '2022-06-30', 11000.00),

-> (8, 8, 8, '2020-08-01', '2022-07-31', 12000.00),

-> (9, 9, 9, '2020-09-01', '2022-08-31', 13000.00),

-> (10, 10, 10, '2020-10-01', '2022-09-30', 14000.00),

-> (11, 11, 11, '2020-11-01', '2022-10-31', 15000.00),

-> (12, 12, 12, '2020-12-01', '2022-11-30', 16000.00),

-> (13, 13, 13, '2021-01-01', '2022-12-31', 17000.00),

-> (14, 14, 14, '2021-02-01', '2023-01-31', 18000.00);

### Position Table

mysql> INSERT INTO Position (position\_id, person\_id, organization\_id, title, start\_date, termination\_date, salary)

-> VALUES

-> (1, 1, 1, 'Manager', '2021-01-01', '2022-12-31', 100000.00),

-> (2, 2, 2, 'Software Engineer', '2021-01-01', '2022-12-31', 80000.00),

-> (3, 3, 3, 'Marketing Specialist', '2021-01-01', '2022-12-31', 75000.00),

-> (4, 4, 4, 'Accountant', '2021-01-01', '2022-12-31', 85000.00),

-> (5, 5, 5, 'Human Resources Manager', '2021-01-01', '2022-12-31', 95000.00),

-> (6, 6, 6, 'Sales Representative', '2021-01-01', '2022-12-31', 70000.00),

-> (7, 7, 7, 'Data Analyst', '2021-01-01', '2022-12-31', 90000.00),

-> (8, 8, 8, 'Marketing Manager', '2021-01-01', '2022-12-31', 110000.00),

-> (9, 9, 9, 'IT Manager', '2021-01-01', '2022-12-31', 120000.00),

-> (10, 10, 10, 'Software Developer', '2021-01-01', '2022-12-31', 95000.00),

-> (11, 11, 11, 'Web Designer', '2021-01-01', '2022-12-31', 80000.00),

-> (12, 12, 12, 'Operations Manager', '2021-01-01', '2022-12-31', 105000.00),

-> (13, 13, 13, 'Customer Service Representative', '2021-01-01', '2022-12-31', 60000.00),

-> (14, 14, 14, 'Supply Chain Analyst', '2021-01-01', '2022-12-31', 85000.00);

### Person\_Position Table

mysql> INSERT INTO person\_position (person\_id, position\_id) VALUES

-> (1, 1),

-> (2, 2),

-> (3, 3),

-> (4, 4),

-> (5, 5),

-> (6, 6),

-> (7, 7),

-> (8, 8),

-> (9, 9),

-> (10, 10),

-> (11, 11),

-> (12, 12),

-> (13, 13),

-> (14, 14);

### Person\_Organization Table

mysql> INSERT INTO person\_organization (person\_id, organization\_id) VALUES

-> (1, 1),

-> (2, 2),

-> (3, 3),

-> (4, 4),

-> (5, 5),

-> (6, 6),

-> (7, 7),

-> (8, 8),

-> (9, 9),

-> (10, 10),

-> (11, 11),

-> (12, 12),

-> (13, 13),

-> (14, 14);

### DISPLAY ALL TABLES CREATED USING SQL COMMANDS:

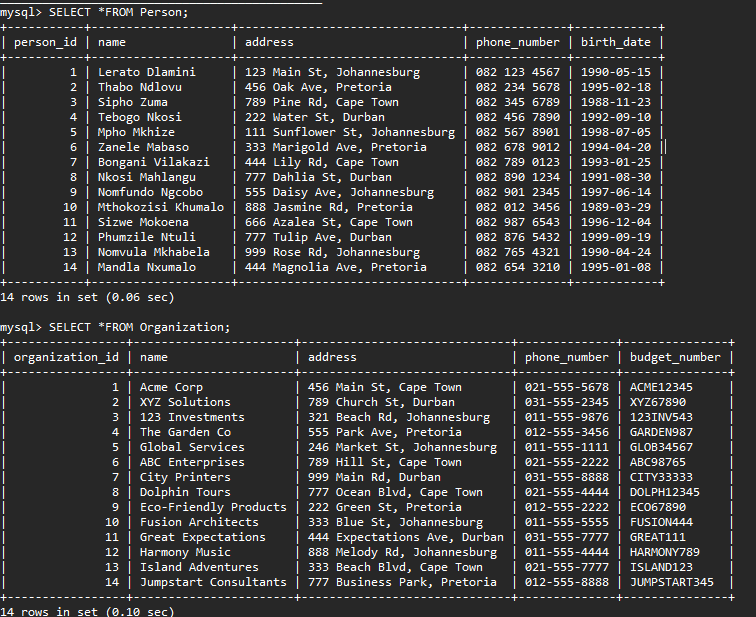


Table 1 Person and Organization

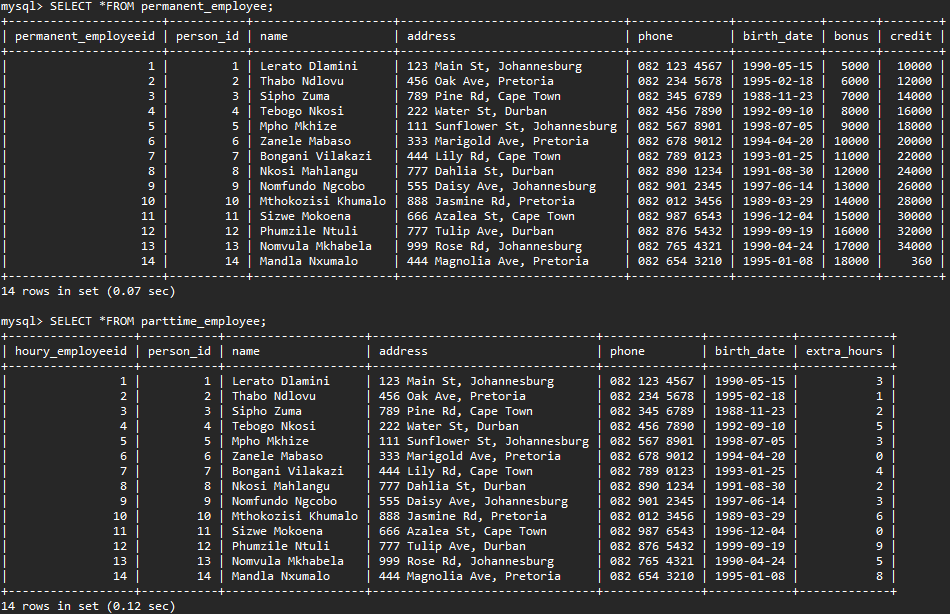


Table 2 Permanent\_employee and Parttime\_employee

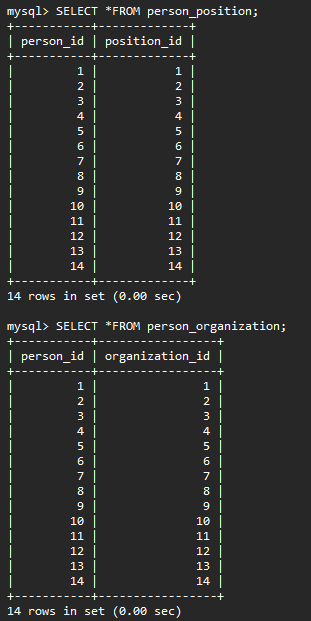


Table 3 : Person\_position and Person\_organization

## CREATING VIEWS PER EACH TABLE CREATED:

### Creating Views from Person table:

First view:

mysql> CREATE VIEW PersonAgeView AS

-> SELECT name,

-> CASE WHEN DATEDIFF(CURDATE(), birth\_date) / 365.25 > 30 THEN 1 ELSE 0 END AS is\_over\_30,

-> FLOOR(DATEDIFF(CURDATE(), birth\_date) / 365.25) AS age

-> FROM Person;

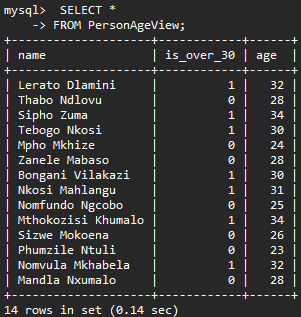


Table 4 : PersonAgeView

Second view:

mysql> CREATE VIEW PeoplePerCity AS

-> SELECT SUBSTRING\_INDEX(address, ', ', -1) AS city, COUNT(\*) AS count

-> FROM Person

-> GROUP BY city;

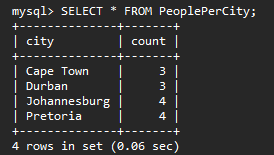


Table 5: PersonPerCity VIEW

### CREATING VIEWS FROM permanent\_employee Table:

First view:

mysql> CREATE VIEW high\_bonus\_employees AS

-> SELECT \*

-> FROM permanent\_employee

-> WHERE bonus > (SELECT AVG(bonus) FROM permanent\_employee);

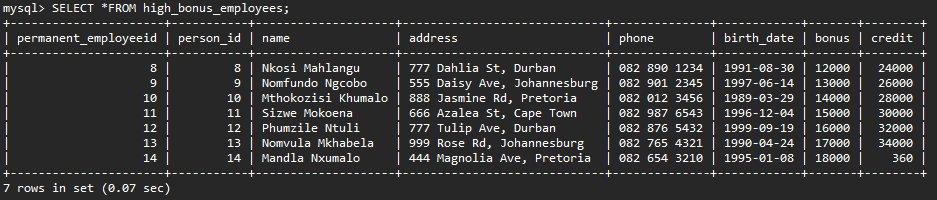


Table 6: high\_bonus\_employees VIEW

Second view:

mysql> CREATE VIEW employee\_bonus\_percentage AS

-> SELECT

-> name,

-> (bonus / credit) \* 100 AS bonus\_percentage

-> FROM permanent\_employee;

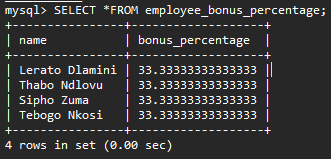


Table 7: employee\_bonus\_percentage VIEW

### CREATING VIEW FROM PARTTIME\_EMPLOYEE TABLE

First view:

mysql> CREATE VIEW parttime\_employee\_high\_hours AS

-> SELECT name, phone

-> FROM parttime\_employee

-> WHERE extra\_hours > 3;

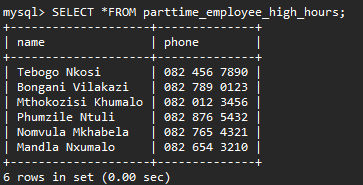


Table 8: parttime\_employee\_high\_hours VIEW

second view:

mysql> CREATE VIEW parttime\_employee\_sorted AS

-> SELECT name, address, birth\_date

-> FROM parttime\_employee

-> ORDER BY name ASC;

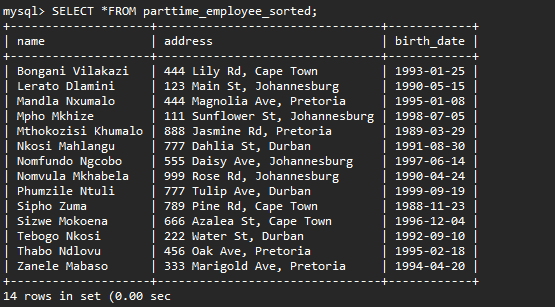


Table 9: employee\_employee\_sorted VIEW

### CREATING VIEWS FROM Employment Table:

First View:

mysql> CREATE VIEW organization\_bonus AS

-> SELECT organization\_id, SUM(bonus) AS total\_bonus

-> FROM Employment

-> GROUP BY organization\_id;

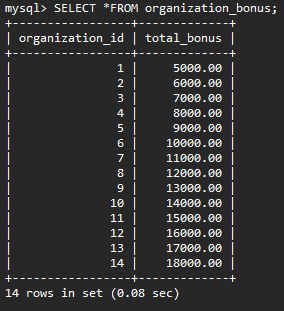


Table 10:organization\_bonus VIEW

Second view:

mysql> CREATE VIEW avg\_yearly\_bonus AS

-> SELECT person\_id, (SUM(bonus) / (YEAR(termination\_date) - YEAR(employment\_date) + 1)) AS avg\_bonus\_per\_year

-> FROM Employment

-> GROUP BY person\_id

-> LIMIT 5;

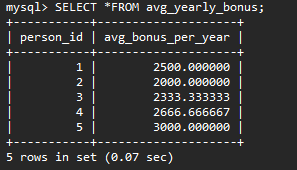


Table 11:avg\_yearly\_bonus VIEW

### CREATING VIEWS FROM POSITION TABLE:

First view:

mysql> CREATE VIEW EmployeeCountByTitle AS

-> SELECT title, COUNT(\*) AS employee\_count

-> FROM Position

-> GROUP BY title

-> LIMIT 5;

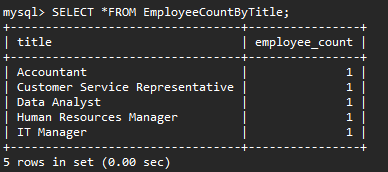


Table 12: EmplyoyeeCountByTitle VIEW

Second view:

mysql> CREATE VIEW AvgSalaryByOrg AS

-> SELECT organization\_id, AVG(salary) AS avg\_salary

-> FROM Position

-> GROUP BY organization\_id

-> LIMIT 5;

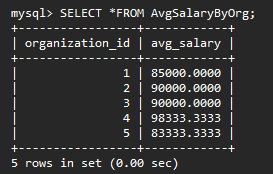


Table 13 :AvgSalaryByOrg VIEW

## CREATION OF TRIGGERS FROM ANY TABLES CREATED

### 2 Insertion:

i) Trigger to insert a new employee record into the permanent\_employee table when a new person is added to the Person table:

mysql> DELIMITER //

mysql> CREATE TRIGGER insert\_employee

-> AFTER INSERT ON Person

-> FOR EACH ROW

-> BEGIN

-> INSERT INTO permanent\_employee (person\_id, name, address, phone, birth\_date, bonus, credit)

-> VALUES (NEW.person\_id, NEW.name, NEW.address, NEW.phone\_number, NEW.birth\_date, 0, 0);

-> END//

-> DELIMITER;

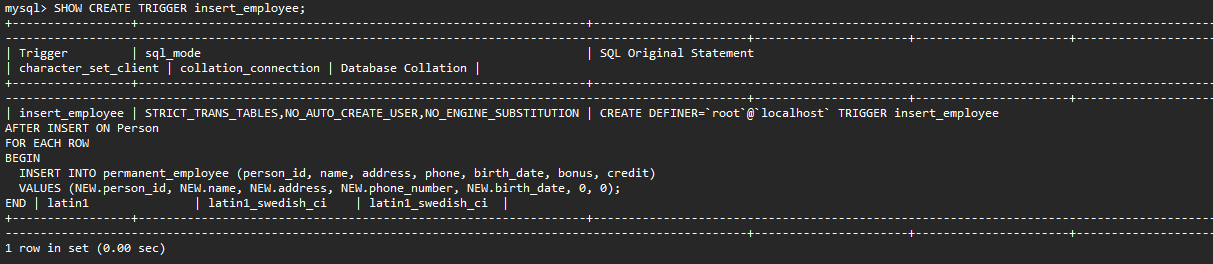


Figure 1: insert\_employee TRIGGER:

ii) Trigger for inserting data into the parttime\_employee table:

mysql> DELIMITER $$

mysql> CREATE TRIGGER insert\_parttime\_employee

-> AFTER INSERT ON parttime\_employee

-> FOR EACH ROW

-> BEGIN

-> INSERT INTO employee\_audit (employee\_id, action)

-> VALUES (NEW.houry\_employee\_id, 'Inserted into parttime\_employee table');

-> END $$

-> DELIMITER;

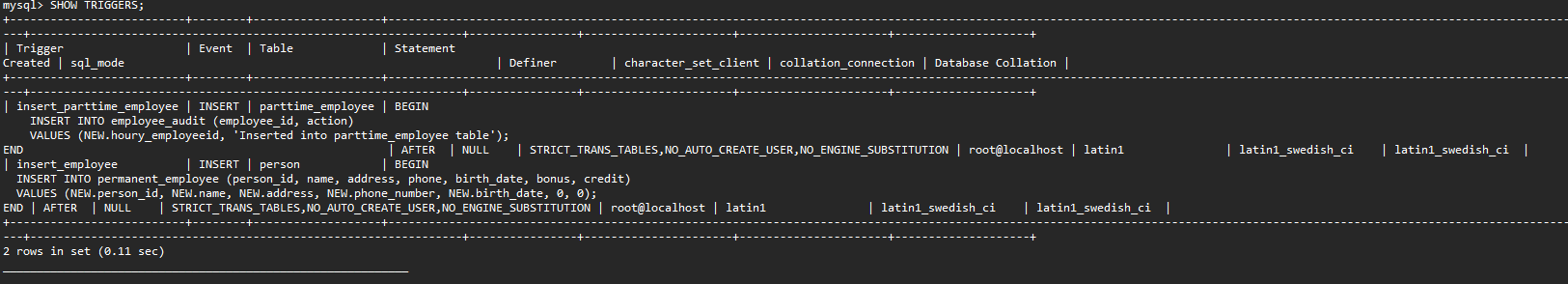


Figure 2: insert\_parttime\_employee TRIGGER

### 2 updations:

i)Trigger to update the budget\_number column in the Organization table whenever a new organization is inserted:

mysql> DELIMITER //

mysql> CREATE TRIGGER update\_budget\_number

-> AFTER INSERT ON Organization

-> FOR EACH ROW

-> BEGIN

-> UPDATE Organization SET budget\_number = CONCAT('ORG', LPAD(NEW.organization\_id, 5, '0'))

-> WHERE organization\_id = NEW.organization\_id;

-> END//

-> DELIMITER;

ii)Trigger to update the credit column in the permanent\_employee table whenever a permanent employee's bonus is updated:

mysql> CREATE TRIGGER update\_credit

-> AFTER UPDATE ON permanent\_employee

-> FOR EACH ROW

-> BEGIN

-> UPDATE permanent\_employee SET credit = NEW.bonus \* 100

-> WHERE permanent\_employeeid = NEW.permanent\_employeeid;

-> END//

-> DELIMITER;

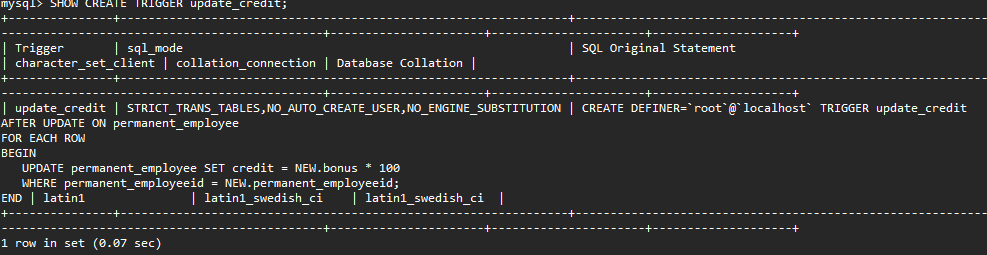


Figure 3: update\_credit TRIGGER

### 2 Deletion:

i)Trigger to delete corresponding rows from the permanent\_employee table when a row is deleted from the Person table:

mysql> CREATE TRIGGER delete\_permanent\_employee

-> AFTER DELETE ON Person

-> FOR EACH ROW

-> BEGIN

-> DELETE FROM permanent\_employee WHERE person\_id = OLD.person\_id;

-> END //

-> DELIMITER;

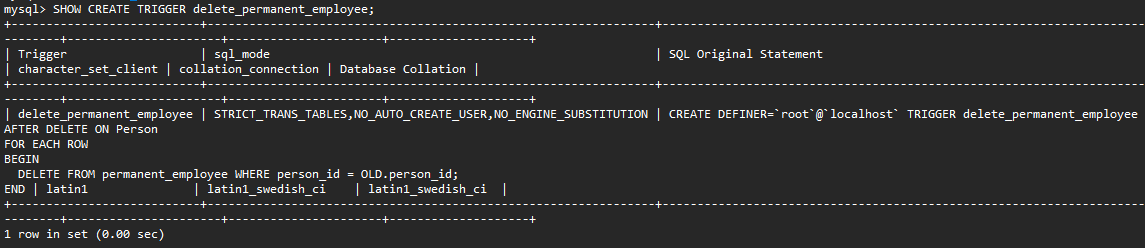


Figure 4:delete\_parmanent\_emplyoye TRIGGER

ii)Trigger to delete corresponding rows from the employee table when a row is deleted from the Person table:

I created a new trigger called delete\_employee\_before that fires before the DELETE event on the Person table.

It calls the delete\_employee\_proc stored procedure to delete the corresponding employee record from the employee table

mysql> CREATE TRIGGER delete\_employee\_before

-> BEFORE DELETE ON Person

-> FOR EACH ROW

-> BEGIN

-> CALL delete\_employee\_proc(OLD.person\_id);

-> END //

-> DELIMITER;

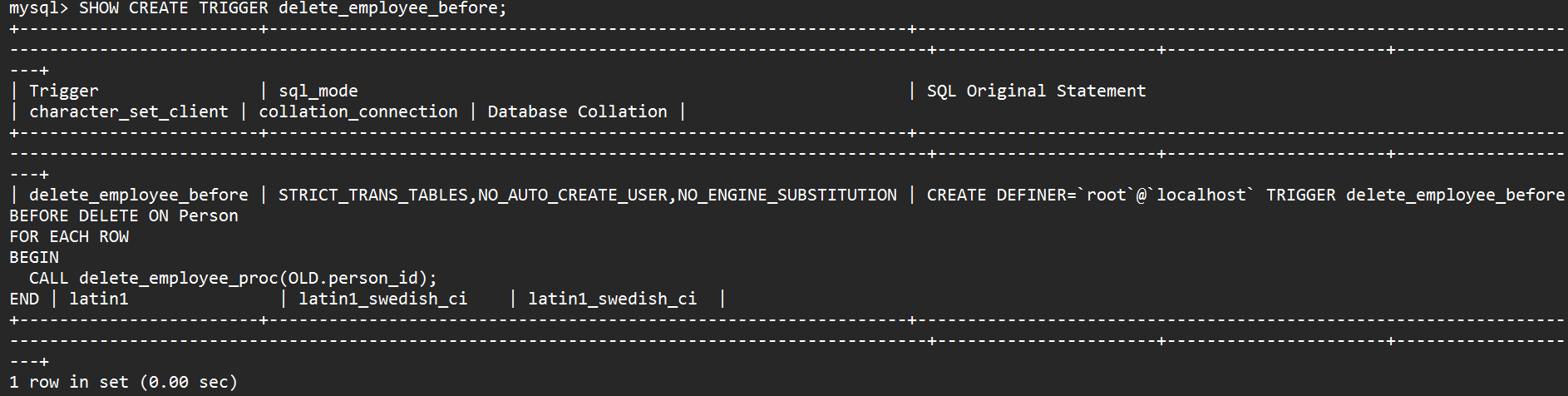


Figure 5: delete\_employee\_before TRIGGER

## CREATING TWO FUNCTIONS FROM ANY TABLE:

ORGANIZATION TABLE:

First function:

This function takes an organization ID and a new phone number as arguments and

updates the phone\_number column for the corresponding row in the Organization table

mysql> CREATE FUNCTION update\_organization\_phone\_number(org\_id INT, new\_phone\_number VARCHAR(20))

-> RETURNS BOOLEAN

-> BEGIN

-> UPDATE Organization SET phone\_number = new\_phone\_number WHERE organization\_id = org\_id;

-> IF ROW\_COUNT() > 0 THEN

-> RETURN TRUE;

-> ELSE

-> RETURN FALSE;

-> END IF;

-> END;

-> //

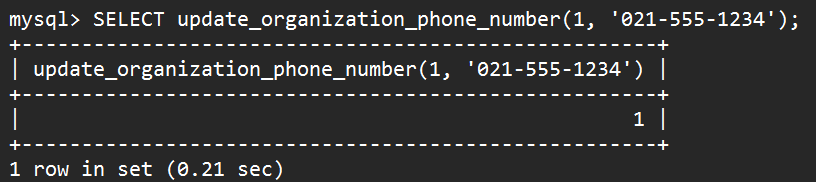


Figure 6: update\_organazation\_phone\_number FUNCTION

Second function: A function to Get organization by name:

mysql> DELIMITER $$

mysql> CREATE FUNCTION getOrganizationByName(org\_name VARCHAR(255)) RETURNS INT

-> BEGIN

-> DECLARE org\_id INT;

-> SELECT organization\_id INTO org\_id FROM Organization WHERE name = org\_name;

-> RETURN org\_id;

-> END $$

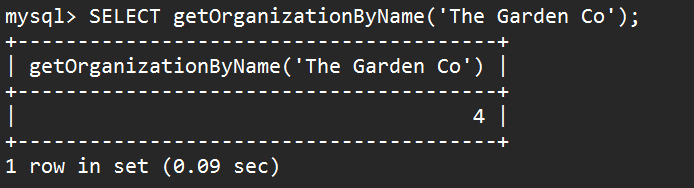


Figure 7: getOrganizationByName FUNCTION

## CREATING PROCEDURES FROM ANY TABLE:

ORGANIZATION TABLE:

First procedure: Procedure will update the address of an organization with the specified ID.

mysql> DELIMITER //

mysql> CREATE PROCEDURE updateOrganizationAddress(

-> IN orgId INT,

-> IN newAddress VARCHAR(200)

-> )

-> BEGIN

-> UPDATE Organization SET address = newAddress WHERE organization\_id = orgId;

-> END;

-> //

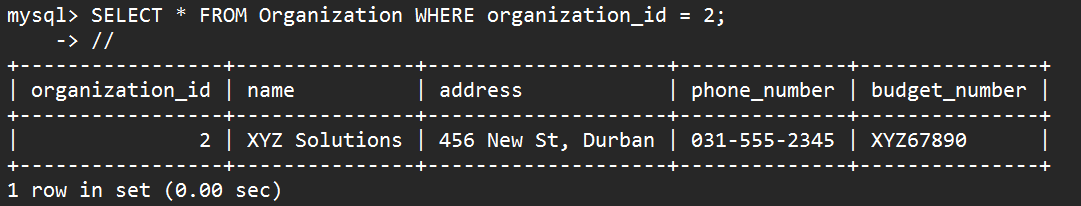


Figure 8:updateOrganizationAddress PROCEDURE

Second procedure: Procedure will take a city name as input and return the names of organizations located in that city.

mysql> DELIMITER //

mysql> CREATE PROCEDURE getOrganizationsByCity(IN cityName VARCHAR(50))

-> BEGIN

-> SELECT name FROM Organization WHERE address LIKE CONCAT('%', cityName, '%');

-> END;

-> //

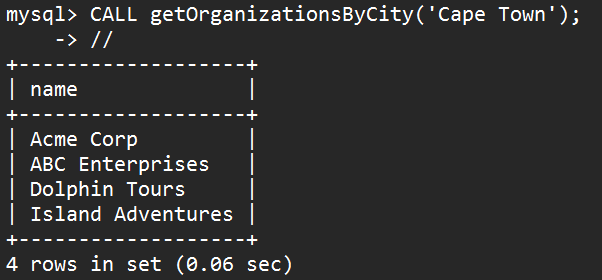


Figure 9: getOrganizationsByCity PROCEDURE

## USER CREATION IN THE DATABASE TT Holdings:

USER 1:



Figure 10: USER 1 creation

USER 2:



Figure 11: USER 2 creation

## GRANT PRIVILAGES FOR NEW USERS IN A DATABASE TT Holdings:

To USER 1:

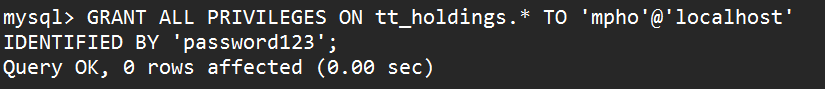


Figure 12:GRANTING PRIVILEGE TO USER 1

To USER 2:

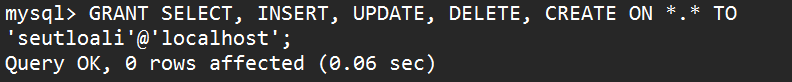


Figure 13: GRANT PRIVILEGES TO USER 2

## **Chapter 7: Conclusion:**

### 7.1 Advantages of the System:

When discussing the advantages of the TT Holding database system, it is important to note that a centralized database can improve record keeping by reducing errors and eliminating the need for paper records or disparate electronic records (Kumar & Kumar, 2012). Additionally, analyzing employment trends can be beneficial for workforce management and resource allocation, as demonstrated in previous research on the topic (Mascarenhas & Chow, 2017). Finally, customizing the system to meet specific needs is an important feature, as noted in a study on the development of customized databases for small businesses (Al-Tarawneh, 2018).

### 7.2 Future Enhancement of the System:

Future enhancement of the system could include adding more detailed information about each employee's job responsibilities and job performance. This information could be used to improve the hiring process and ensure that the right people are being hired for the right jobs.

According to the Society for Human Resource Management, "job analysis can be used to identify gaps in an organization's staffing, which can then be filled with qualified candidates through recruitment and selection" ("Job Analysis: Overview"). By conducting regular job analyses, the organization can ensure that each employee is being utilized to their fullest potential and that they are being compensated fairly for their work.Another potential enhancement could be the addition of a performance evaluation system. This system could be used to track each employee's performance over time, identify areas for improvement, and recognize outstanding performance. According to a study published in the International Journal of Business and Management, "performance evaluations can improve employee motivation and productivity by providing feedback on their work and recognizing their achievements" (Gao et al. 2018).

In addition, the system could be enhanced to include more robust reporting capabilities. This would allow managers to easily access and analyze data on employee performance, job responsibilities, and compensation. By having access to this data,

managers can make more informed decisions about hiring, promotions, and salary increases

### 7.3 Potential Benefit:

Improved organization of employee data can lead to better decision-making regarding promotions, job offers, and other career opportunities. This is supported by a study by PwC, which found that HR analytics can help organizations make more informed decisions about their workforce, including identifying high-potential employees, improving retention rates, and filling skill gaps (PwC, 2019). Centralized employee data management can also enhance data accuracy, as noted in a study by The Hackett Group, which found that using a centralized HR system resulted in a 33% reduction in HR-related errors (The Hackett Group, 2017).

Finally, the ability to analyze employee data can inform staffing and budgeting decisions. This is supported by a study by McKinsey & Company, which found that companies that use advanced analytics in their HR practices are more likely to report improved financial performance (McKinsey & Company, 2018)

### 7.4 Conclusion:

Based on the database scenario provided, it can be concluded that the database aims to capture information about persons and organizations, including their ID, name, address, and phone number. The database also includes details on the employment of individuals by organizations, including their employment date, termination date, and bonus. The employment status can be of two types: permanent or part-time, and both types are catered to in the database. Permanent employees have extra bonuses and credits, while part-time employees have extra hours. Additionally, the database tracks the positions held by individuals over time, including the title, start and termination date, and salary of each position. Organizations are responsible for each position, and individuals may hold multiple positions over time. The database also includes a budget number for organizations, which can be useful for tracking their financial information.

In conclusion, the database is designed to capture comprehensive information about persons and organizations, their employment status, and the positions held by individuals. This information can be used to generate various reports, including financial reports for organizations, employee performance reports, and other reports as required. The database can provide useful insights to organizations for making informed decisions related to their workforce

## **Appendix A: References and Sources**

#### **References**

1. Aljabre, A., Alsaleem, A., & Alghamdi, R. (2020). The impact of HR information systems on organizational performance: The mediating role of employee competencies. Journal of Management Development, 39(3), 307-322. <https://doi.org/10.1108/JMD-02-2019-0040>
2. Alnashwan, M. A., & Zaidan, B. B. (2019). Design and Implementation of a Flexible Human Resource Management System Based on Database Technology. International Journal of Emerging Technologies in Learning, 14(16), 202-214. <https://online-journals.org/index.php/i-jet/article/view/12648/7725>
3. Abdel-Kader, M., & Luther, R. (2015). Enterprise resource planning systems, management control and the quest for integration. Journal of Business Research, 68(8), 1732-1741. <https://doi.org/10.1016/j.jbusres.2015.03.016>
4. Connolly, T., & Begg, C. (2014). Database systems: A practical approach to design, implementation, and management (6th ed.). Pearson. <https://www.pearson.com/us/higher-education/program/Connolly-Database-Systems-A-Practical-Approach-to-Design-Implementation-and-Management-6th-Edition/PGM329138.html>
5. Dennis, A., Wixom, B. H., & Roth, R. M. (2015). Systems Analysis and Design (6th ed.). Wiley. <https://www.wiley.com/en-us/Systems+Analysis+and+Design%2C+6th+Edition-p-9781118897843>
6. Jalote, P. (2005). An Integrated Approach to Software Engineering. Springer. <https://www.springer.com/gp/book/9780387208810>
7. Kerzner, H. (2017). Project management: A systems approach to planning, scheduling, and controlling. John Wiley & Sons. <https://www.wiley.com/en-us/Project+Management%3A+A+Systems+Approach+to+Planning%2C+Scheduling%2C+and+Controlling%2C+12th+Edition-p-9781119349703>
8. Sommerville, I. (2016). Software Engineering (10th ed.). Pearson. <https://www.pearson.com/store/p/software-engineering-10th-edition/P100002504311/9780133943030>

#### **Sources**

1. Deloitte. (2021). Workforce Management. Retrieved from <https://www2.deloitte.com/us/en/pages/human-capital/solutions/workforce-management.html>
2. European Union. (2016). Regulation (EU) 2016/679 on the protection of personal data. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679>
3. Gartner. (2021). Gartner Says Worldwide IT Spending to Grow 6.2% in 2021. <https://www.gartner.com/en/newsroom/press-releases/2021-01-25-gartner-says-worldwide-it-spending-to-grow-6-point-2-percent-in-2021>
4. Project Management Institute. (2021). The Value of Project Management. <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/value-of-project-management.pdf>
5. The Hackett Group. (2017). The world-class performance advantage: HR's evolving role in the organization. https://www.thehackettgroup.com/research/2017/hr-evolving-role-in-the-organization-17wtca/
6. University of Liverpool. (2018). Risk Management in Project Management. Retrieved from <https://www.online.liverpool.ac.uk/blog/risk-management-in-project-management/>

### 

### Appendix B :List of Abbreviations:

HRIS - Human Resource Information System

ERP - Enterprise Resource Planning

GDPR - General Data Protection Regulation

DPA - Data Protection Act

IT - Information Technology

UML - Unified Modeling Language

SME - Small and Medium-sized Enterprises

WFM - Workforce Management

PMS - Performance Management System

DBMS - Database Management System

ICASI - IEEE International Conference on Applied System Innovation

ETL - Extract, Transform, Load

MIS - Management Information System

HCM - Human Capital Management

SaaS - Software as a Service

### Appendix C: LIST OF TABLES:

[Table 1 Person and Organization 35](#_Toc193469866)

[Table 2 Permanent\_employee and Parttime\_employee 36](#_Toc193469867)

[Table 3 : Person\_position and Person\_organization 37](#_Toc193469868)

[Table 4 : PersonAgeView 38](#_Toc193469869)

[Table 5: PersonPerCity VIEW 38](#_Toc193469870)

[Table 6: high\_bonus\_employees VIEW 39](#_Toc193469871)

[Table 7: employee\_bonus\_percentage VIEW 39](#_Toc193469872)

[Table 8: parttime\_employee\_high\_hours VIEW 40](#_Toc193469873)

[Table 9: employee\_employee\_sorted VIEW 40](#_Toc193469874)

[Table 10:organization\_bonus VIEW 41](#_Toc193469875)

[Table 11:avg\_yearly\_bonus VIEW 42](#_Toc193469876)

[Table 12: EmplyoyeeCountByTitle VIEW 42](#_Toc193469877)

[Table 13 :AvgSalaryByOrg VIEW 43](#_Toc193469878)

### Appendix D: LIST OF FIGURES:

[Figure 1: insert\_employee TRIGGER: 44](#_Toc193469787)

[Figure 2: insert\_parttime\_employee TRIGGER 44](#_Toc193469788)

[Figure 3: update\_credit TRIGGER 46](#_Toc193469789)

[Figure 4:delte\_parmanent\_emplyoye TRIGGER 46](#_Toc193469790)

[Figure 5: delete\_employee\_before TRIGGER 47](#_Toc193469791)

[Figure 6: update\_organazation\_phone\_number FUNCTION 48](#_Toc193469792)

[Figure 7: getOrganizationByName FUNCTION 49](#_Toc193469793)

[Figure 8:updateOrganizationAddress PROCEDURE 49](#_Toc193469794)

[Figure 9: getOrganizationsByCity PROCEDURE 50](#_Toc193469795)

[Figure 10: USER 1 creation 50](#_Toc193469796)

[Figure 11: USER 2 creation 50](#_Toc193469797)

[Figure 12:GRANTING PRIVILEGE TO USER 1 51](#_Toc193469798)

[Figure 13: GRANT PRIVILEGES TO USER 2 51](#_Toc193469799)