# Report on Database Design of TT Holding System

Table of Contents

Chapter 1: Introduction  
1.1 Problem Statement 4  
1.2 Problem Solving 4  
1.3 Objective 4  
1.4 Scope & Constraint 4

Chapter 2: Literature Review  
2.1 Introduction 5  
2.2 Reviewing the Literature 5  
2.3 Findings & Discussion 5

Chapter 3: Methodology  
3.1 Requirement Analysis 5  
3.2 System Design 5  
3.2.1 Architectural Design 5  
3.2.2 UML Diagrams 5  
3.3 System Implementation / Prototyping 5-6  
3.4 Testing 6

Chapter 4: System Initiation and Planning 6  
4.1 Assessing Project Feasibility 6  
4.2 Project Plan 6

Chapter 5: System Analysis 6  
5.1 Determining System Requirements 6  
5.2 Structuring System Requirements 6

Chapter 6: Conclusion 6  
6.1 Advantages of the System 6  
6.2 Future Enhancement of the System 6-7  
6.3 Potential Benefit 7  
6.4 Conclusion 7

References 7

Appendices 7-29

List of Figures

Figure 1: Database Schema Diagram

Figure 2: UML Class Diagram

List of Tables

Table 1: PERSON Table Structure

Table 2: ORGANISATION Table Structure

Table 3: EMPLOYMENT Table Structure

Table 4: Position Table Structure

Table 5: Part-Time Employee Table Structure

Table 6: Permanent Employee Table Structure

Table 7: Internal Organisation Table Structure

Table 8: External Organisation Table Structure

List of Abbreviations

DBMS: Database Management System

UML: Unified Modeling Language

INT: Integer

DECIMAL: Decimal Number

Abstract

This report presents the design and implementation of a relational database for tt\_holding, which manages personnel and organizational data. The database consists of multiple tables, including PERSON, ORGANISATION, EMPLOYMENT, Position, Part\_Time\_Employee, Permanent\_Employee, Internal\_Organisation, and External\_Organisation. The report outlines the problem statement, objectives, methodology, and findings, providing a comprehensive overview of the database structure and its functionalities.

Chapter 1: Introduction

* 1. Problem Statement

TT Holdings, an enterprise, manages various entities such as employment, people, organizations, positions, and more. The challenge lies in effectively structuring the data to ensure that each employee's information is accurate and separate from other records. This involves establishing clear relationships between tables to avoid data redundancy while ensuring that data across different tables works cohesively together. The goal is to create a well-organized database schema that maintains data integrity and facilitates efficient data retrieval and reporting.A company is establishing a system design that will provide.

* 1. Problem Solving

The solution involves designing a structured database system that accurately identifies and defines all relevant entities, attributes, and relationships within the construction of TT Holdings. This includes creating tables for each entity—such as employees, organizations, positions, and employment records—while ensuring that relationships between these tables are clearly defined to facilitate data integrity and minimize redundancy. By implementing a relational database design, we can ensure that data is organized in a way that allows for efficient querying and reporting, enabling seamless interaction between related data points and supporting the overall operational needs of the enterprise.

1.3 Objective

The primary objective of this project is to develop a robust and well-structured relational database that ensures the accurate storage and retrieval of data. This database will serve as a functional management system, enabling efficient tracking of employee information, organizational relationships, and employment records. By implementing best practices in database design, the project aims to enhance data integrity, minimize redundancy, and facilitate seamless interactions between related entities, ultimately supporting the operational needs of TT Holdings.

1.4 Scope & Constraint

The scope of this project includes the design and creation of a relational database system that encompasses the identification of key entities such as employees, organizations, positions, and employment records, along with their attributes and interrelationships. It involves implementing the database schema in a suitable database management system, populating it with accurate initial data, and providing basic querying capabilities for effective data retrieval and manipulation.

Chapter 2: Literature Review

2.1 Introduction

The literature review focuses on existing database management systems and their applications in personnel management. It highlights the importance of structured data storage and retrieval.

2.2 Reviewing the Literature

Recent studies emphasize the significance of relational databases in managing complex data relationships, particularly in human resource management. Various publications discuss best practices in database design, normalization, and data integrity.

2.3 Findings & Discussion

The findings indicate that a well-structured relational database can significantly enhance data management efficiency, reduce redundancy, and improve data accuracy.

Chapter 3: Methodology

3.1 Requirement Analysis

The requirements for the database were provided by TT Holding enterprise to design a database system that has two user with each user having two grant privileges over the database.

3.2 System Design

3.2.1 Architectural Design

The database consists of several interconnected tables, each serving a specific purpose. The main tables include:

PERSON: Stores individual data.

ORGANISATION: Stores organizational data.

EMPLOYMENT: Links employees to organizations.

Position: Details job positions within organizations.

Part\_Time\_Employee: Stores data for part-time employees.

Permanent\_Employee: Stores data for permanent employees.

Internal\_Organisation: Contains data for internal departments.

External\_Organisation: Contains data for external partnerships.

3.3 System Implementation / Prototyping

The implementation phase involved creating the database schema, defining relationships between tables, and inserting initial data. Prototyping was conducted to ensure that the database meets the user requirements and functioned as intended.

3.4 Testing

Testing was conducted for the functionality of all tables, functions, triggers and user authentication. The relational database works perfectly towards its intended use and requirements needs.

Chapter 4: System Initiation and Planning

4.1 Assessing Project Feasibility

A feasibility study was conducted to evaluate the technical, operational, and financial aspects of the project. The study confirmed that the project was viable and aligned with organizational goals.

4.2 Project Plan

The blueprint is to design a layout of all entities with each having their own attribute providing defined relationships, then create a database extracted from the blueprint.

Chapter 5: System Analysis

5.1 Determining System Requirements

System requirements were identified through stakeholder interviews and analysis of existing processes. Key requirements included data storage, retrieval capabilities, and reporting functionalities.

5.2 Structuring System Requirements

The requirements were structured into functional and non-functional categories, ensuring clarity and alignment with project objectives.

Chapter 6: Conclusion

6.1 Advantages of the System

The advantages implemented database, including

* Improved data management
* Reduced redundancy
* Enhanced reporting capabilities
* User Authentication

6.2 Future Enhancement of the System

The system in the future should develop a proper functionality on its employees in relation to the employee type and what job description they would doing and again the system should have a proper enhancement with more users using the system but several privileges

6.3 Potential Benefit

The database system is expected to provide more data usage on any entity that is inducted into the system for its functional usage prior other entities.

6.4 Conclusion

In conclusion, the design and implementation of the tt\_holding database provides a structured interaction of data working within the enterprise and providing accurate results over time.

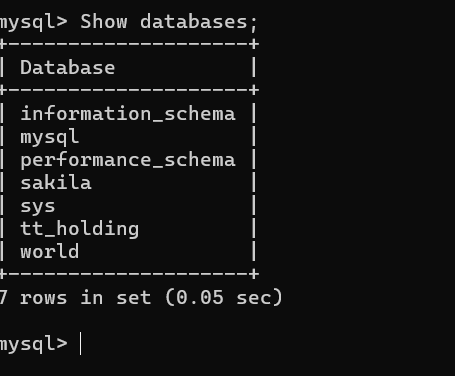
Chapter 7: References

Hoffer, J., Venkataraman, R. and Topi, H. (2016). Chapter 7: Enhanced ERD. In: \*Modern database management\*. 12th ed. Donna Battista, pp. 150-162.

Coronel, C., Morris, S. and Rob, P. (2011). \*Database systems: design, implementation, and management\*. 9th ed. Joe Sabatino.

Chapter 8: Appendices

Creation of Database

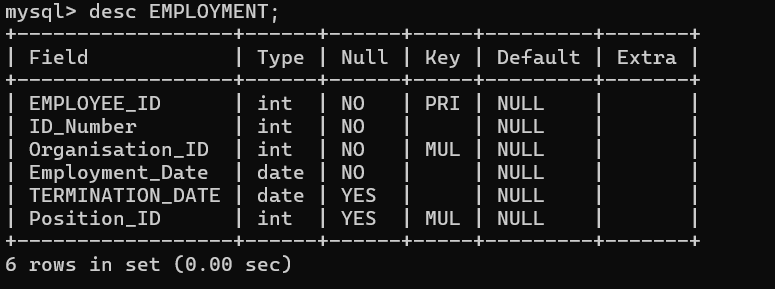


CREATE DATABASE tt\_holding;

use tt\_holding;

Tables Creation

1. Employment



CREATE TABLE EMPLOYMENT (

EMPLOYEE\_ID INT(15) PRIMARY KEY NOT NULL,

ID\_Number INT NOT NULL,

Organisation\_ID INT(15) NOT NULL,

Employment\_Date DATE NOT NULL,

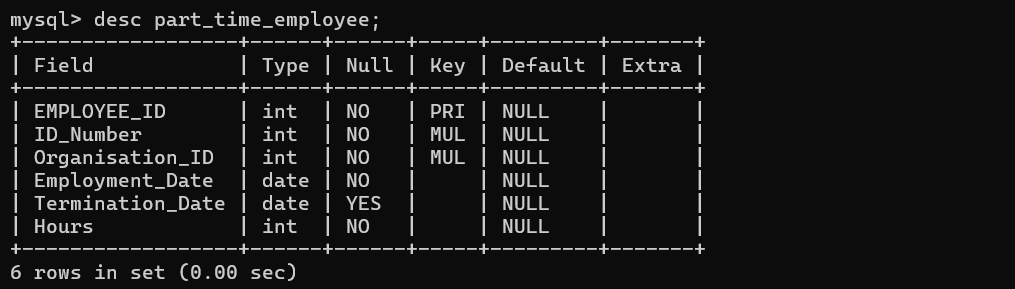
TERMINATION\_DATE DATE,

Position\_ID INT(15),

FOREIGN KEY (Position\_ID) REFERENCES Position(Position\_ID)

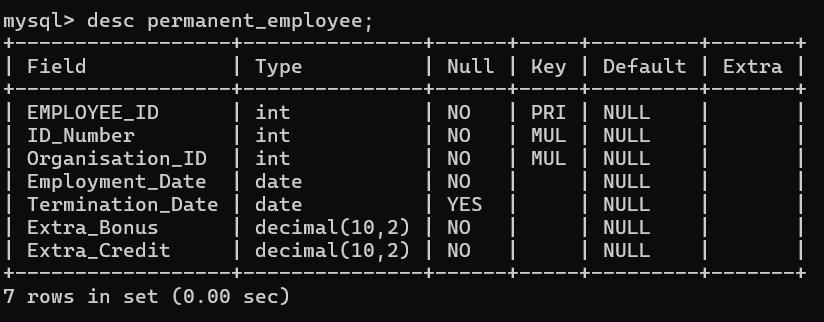
);

1. Part Time Employee



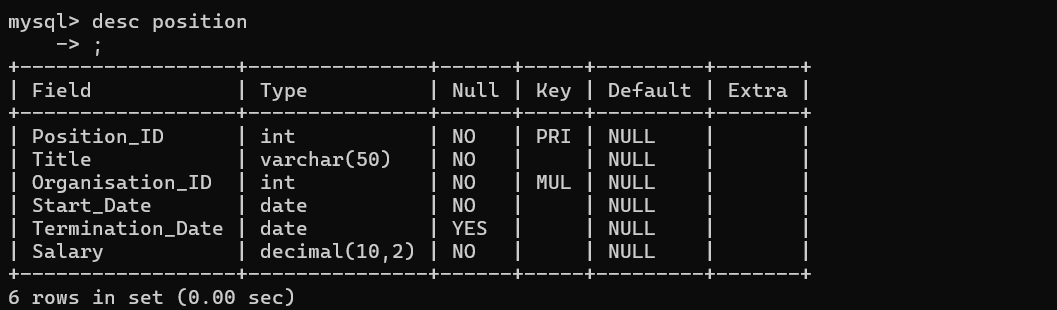
CREATE TABLE Part\_Time\_Employee (EMPLOYEE\_ID INT(15) PRIMARY KEY NOT NULL, ID\_Number INT NOT NULL, Organisation\_ID INT(15) NOT NULL, Employment\_Date DATE NOT NULL, Termination\_Date DATE, Hours INT NOT NULL, FOREIGN KEY (ID\_Number) REFERENCES PERSON(ID\_Number), FOREIGN KEY (Organisation\_ID) REFERENCES ORGANISATION(Organisation\_ID));

1. Permanent Employee



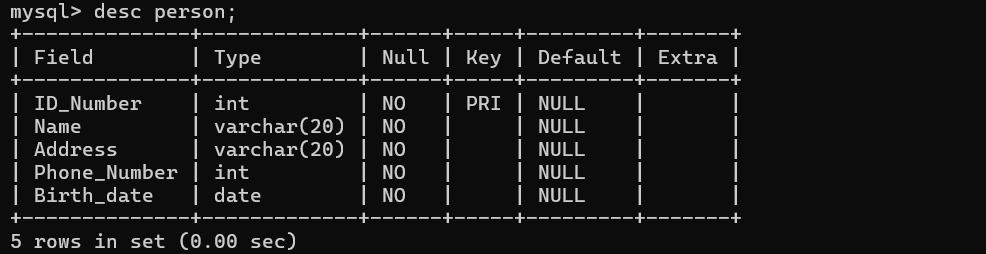
CREATE TABLE Permanent\_Employee (EMPLOYEE\_ID INT(15) PRIMARY KEY NOT NULL, ID\_Number INT NOT NULL, Organisation\_ID INT(15) NOT NULL, Employment\_Date DATE NOT NULL, Termination\_Date DATE, Extra\_Bonus DECIMAL(10, 2) NOT NULL, Extra\_Credit DECIMAL(10, 2) NOT NULL, FOREIGN KEY (ID\_Number) REFERENCES PERSON(ID\_Number), FOREIGN KEY (Organisation\_ID) REFERENCES ORGANISATION(Organisation\_ID));

1. Position



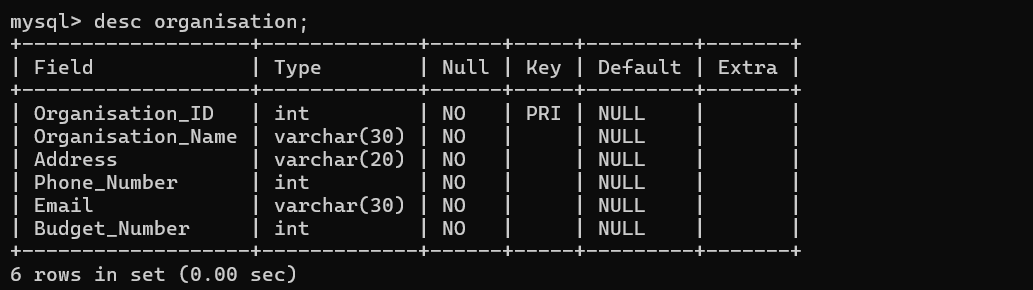
CREATE TABLE Position (Position\_ID INT(15) PRIMARY KEY NOT NULL, Title VARCHAR(50) NOT NULL, Organisation\_ID INT(15) NOT NULL, Start\_Date DATE NOT NULL, Termination\_Date DATE, Salary DECIMAL(10, 2) NOT NULL, FOREIGN KEY (Organisation\_ID) REFERENCES ORGANISATION(Organisation\_ID));

1. Person



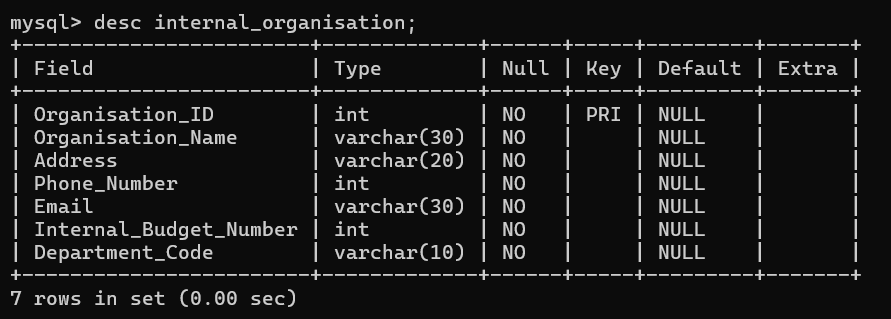
CREATE TABLE PERSON (ID\_Number INT(15) PRIMARY KEY NOT NULL, Name VARCHAR(20) NOT NULL, Address VARCHAR(20) NOT NULL, Phone\_Number INT(8) NOT NULL, Birth\_date DATE NOT NULL);

1. Organisation



CREATE TABLE ORGANISATION (Organisation\_ID INT(15) PRIMARY KEY NOT NULL, Organisation\_Name VARCHAR(30) NOT NULL, Address VARCHAR(20) NOT NULL, Phone\_Number INT(8) NOT NULL, Email VARCHAR(30) NOT NULL, Budget\_Number INT(4) NOT NULL);

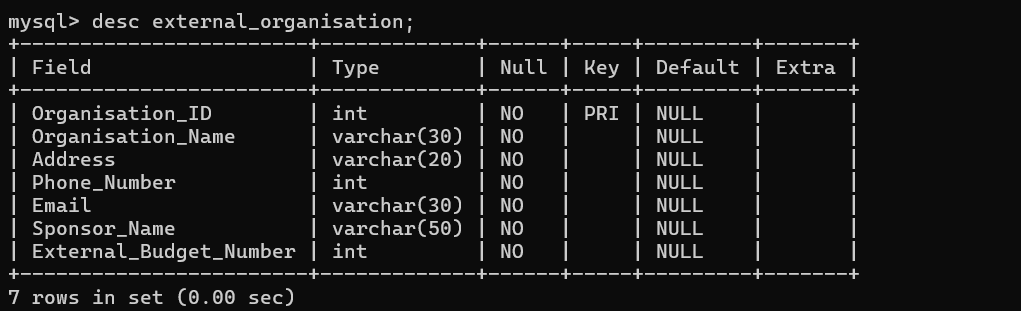
1. Internal Organisation



CREATE TABLE Internal\_Organisation (Organisation\_ID INT(15) PRIMARY KEY NOT NULL, Organisation\_Name VARCHAR(30) NOT NULL, Address VARCHAR(20) NOT NULL, Phone\_Number INT(8) NOT NULL, Email VARCHAR(30) NOT NULL, Internal\_Budget\_Number INT(4) NOT NULL, Department\_Code VARCHAR(10) NOT NULL,

FOREIGN KEY (Organisation\_ID) REFERENCES ORGANISATION(Organisation\_ID));

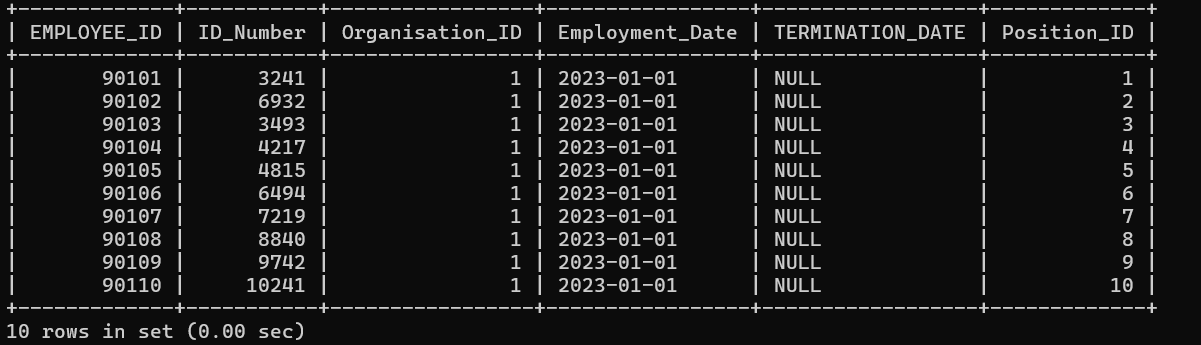
1. External Organisation



CREATE TABLE External\_Organisation (Organisation\_ID INT(15) PRIMARY KEY NOT NULL, Organisation\_Name VARCHAR(30) NOT NULL, Address VARCHAR(20) NOT NULL, Phone\_Number INT(8) NOT NULL, Email VARCHAR(30) NOT NULL, Sponsor\_Name VARCHAR(50) NOT NULL, External\_Budget\_Number INT(4) NOT NULL, FOREIGN KEY (Organisation\_ID) REFERENCES ORGANISATION(Organisation\_ID));

# Data Insertion

# Employement



INSERT INTO EMPLOYMENT (EMPLOYEE\_ID, ID\_Number, Organisation\_ID, Employment\_Date, TERMINATION\_DATE, Position\_ID) VALUES

(90101, 003241, 1, '2023-01-01', NULL, '0001'),

(90102, 006932, 1, '2023-01-01', NULL, '0002'),

(90103, 003493, 1, '2023-01-01', NULL, '0003'),

(90104, 004217, 1, '2023-01-01', NULL, '0004'),

(90105, 004815, 1, '2023-01-01', NULL, '0005'),

(90106, 006494, 1, '2023-01-01', NULL, '0006'),

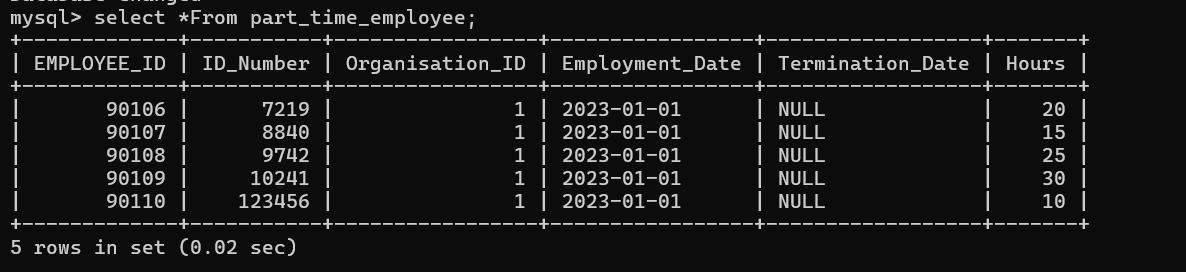
(90107, 007219, 1, '2023-01-01', NULL, '0007'),

(90108, 008840, 1, '2023-01-01', NULL, '0008'),

(90109, 009742, 1, '2023-01-01', NULL, '0009'),

(90110, 010241, 1, '2023-01-01', NULL, '0010');

# Part Time Employee



INSERT INTO Part\_Time\_Employee (EMPLOYEE\_ID, ID\_Number, Organisation\_ID, Employment\_Date, Termination\_Date, Hours) VALUES

(90106, 007219, 1, '2023-01-01', NULL, 20),

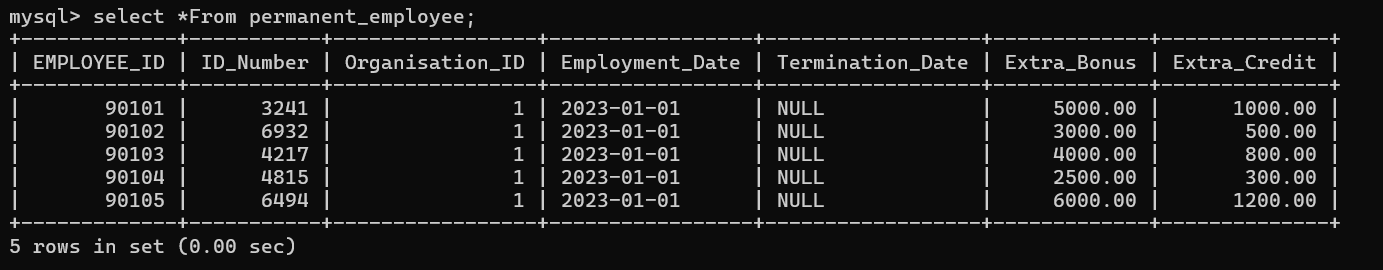
(90107, 008840, 1, '2023-01-01', NULL, 15),

(90108, 009742, 1, '2023-01-01', NULL, 25),

(90109, 010241, 1, '2023-01-01', NULL, 30),

(90110, 123456, 1, '2023-01-01', NULL, 10);

# Permanent Employee



INSERT INTO Permanent\_Employee (EMPLOYEE\_ID, ID\_Number, Organisation\_ID, Employment\_Date, Termination\_Date, Extra\_Bonus, Extra\_Credit) VALUES

(90101, 003241, 1, '2023-01-01', NULL, 5000.00, 1000.00),

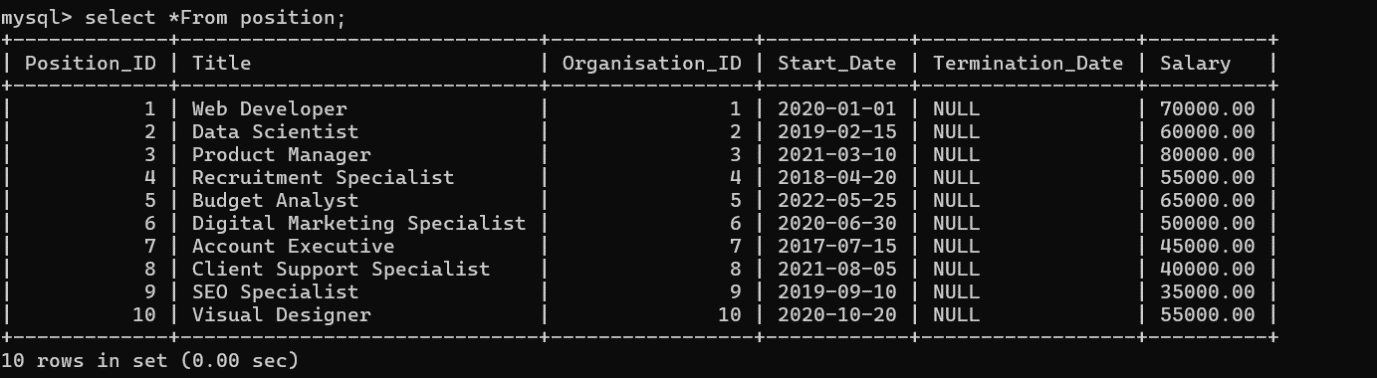
(90102, 006932, 1, '2023-01-01', NULL, 3000.00, 500.00),

(90103, 004217, 1, '2023-01-01', NULL, 4000.00, 800.00),

(90104, 004815, 1, '2023-01-01', NULL, 2500.00, 300.00),

(90105, 006494, 1, '2023-01-01', NULL, 6000.00, 1200.00);

# Position



INSERT INTO Position (Position\_ID, Title, Organisation\_ID, Start\_Date, Termination\_Date, Salary) VALUES

('0001', 'Web Developer', 1, '2020-01-01', NULL, 70000.00),

('0002', 'Data Scientist', 2, '2019-02-15', NULL, 60000.00),

('0003', 'Product Manager', 3, '2021-03-10', NULL, 80000.00),

('0004', 'Recruitment Specialist', 4, '2018-04-20', NULL, 55000.00),

('0005', 'Budget Analyst', 5, '2022-05-25', NULL, 65000.00),

('0006', 'Digital Marketing Specialist', 6, '2020-06-30', NULL, 50000.00),

('0007', 'Account Executive', 7, '2017-07-15', NULL, 45000.00),

('0008', 'Client Support Specialist', 8, '2021-08-05', NULL, 40000.00),

('0009', 'SEO Specialist', 9, '2019-09-10', NULL, 35000.00),

('0010', 'Visual Designer', 10, '2020-10-20', NULL, 55000.00);

# Person



INSERT INTO PERSON (ID\_Number, Name, Address, Phone\_Number, Birth\_date) VALUES

(003241, 'Alina Sello', '123 kingsway St', 62345678, '2001-01-15'),

(006932, 'Lineo Rapene', '456 Moshoeshoe I St', 53456789, '2000-02-20'),

(003493, 'Qilo Nnete', '789 Florida St', 54567890, '1999-03-25'),

(004217, 'Bobile Itekeng', '321 Ha Thamae', 65678901, '2005-04-30'),

(004815, 'Itumeleng Gladys', '654 Lesia', 56789012, '1995-05-05'),

(006494, 'Sehobela Khauta', '987 Motimposo', 67890123, '2004-06-10'),

(007219, 'Limpho Moreki', '159 Thabong', 68901234, '1998-07-15'),

(008840, 'Tsebeletso Monki', '753 Lithabaneng', 59012345, '1996-08-20'),

(009742, 'Thabo Moses', '852 Thetsane', 60123456, '2002-09-25'),

(010241, 'Julia Mosola', '369 TonaKholo St', 51234567, '2003-10-30');

(123456, 'Thabo Mokoena', 'Maseru', 64567901, '1985-01-15'),

(234567, 'Lerato Nkosi', 'Johannesburg', 56789012, '1990-02-20'),

(345678, 'Sipho Khumalo', 'Bloemfontein', 64567891, '1988-03-25'),

(456789, 'Nandi Dlamini', 'Pretoria', 56789002, '1992-04-30'),

(567890, 'Mpho Ramaphosa', 'Maseru', 64589013, '1987-05-05'),

(678901, 'Palesa Mothibi', 'Durban', 67890124, '1993-06-10'),

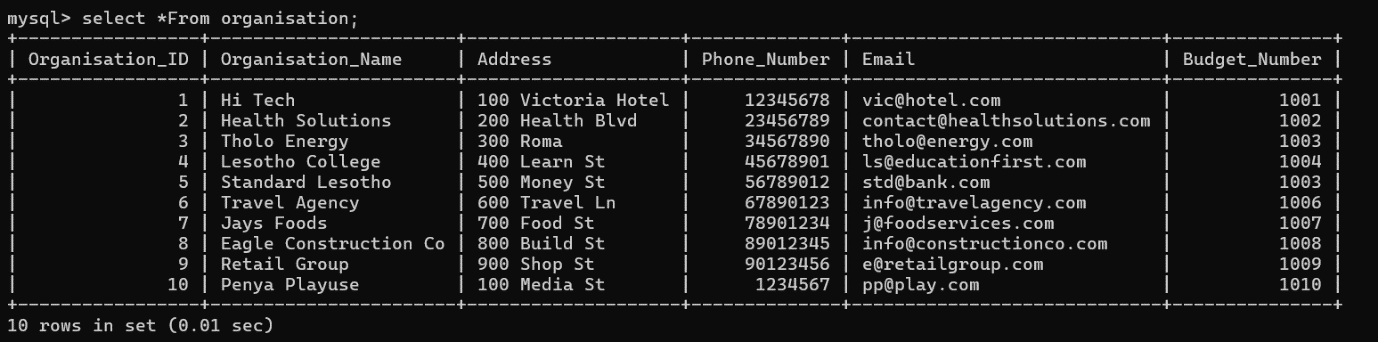
(789012, 'Kabelo Matlosa', 'Maseru', 59012345, '1980-07-15'),

(890123, 'Tshepo Mofolo', 'Johannesburg', 67812346, '1984-08-20'),

(901234, 'Bontle Mofolo', 'Maseru', 64523457, '1991-09-25'),

(012345, 'Dineo Makhura', 'Pretoria', 51234568, '1986-10-30');

# Organisation



INSERT INTO ORGANISATION (Organisation\_ID, Organisation\_Name, Address, Phone\_Number, Email, Budget\_Number) VALUES

(1, 'Hi Tech', '100 Victoria Hotel', 12345678, 'vic@hotel.com', 1001),

(2, 'Health Solutions', '200 Health Blvd', 23456789, 'contact@healthsolutions.com', 1002),

(3, 'Tholo Energy', '300 Roma', 34567890, 'tholo@energy.com', 1003),

(4, 'Lesotho College', '400 Learn St', 45678901, 'ls@educationfirst.com', 1004),

(5, 'Standard Lesotho', '500 Money St', 56789012, 'std@bank.com', 1003),

(6, 'Travel Agency', '600 Travel Ln', 67890123, 'info@travelagency.com', 1006),

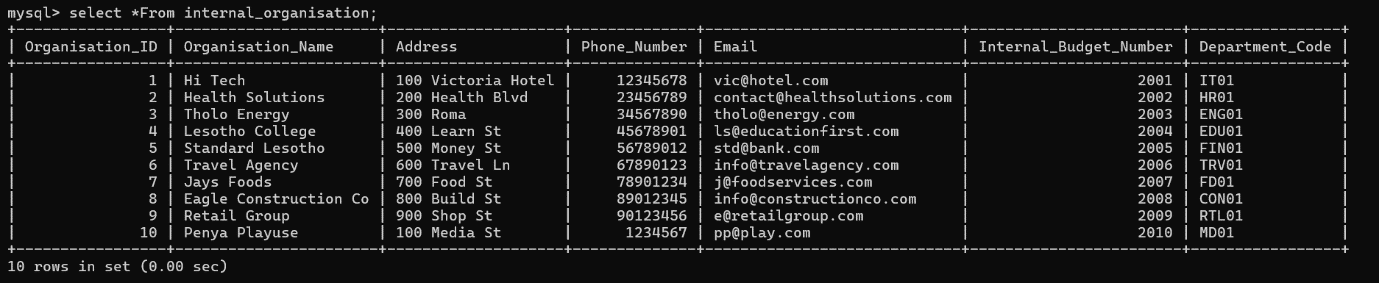
(7, 'Jays Foods', '700 Food St', 78901234, 'j@foodservices.com', 1007),

(8, 'Eagle Construction Co', '800 Build St', 89012345, 'info@constructionco.com', 1008),

(9, 'Retail Group', '900 Shop St', 90123456, 'e@retailgroup.com', 1009),

(10, 'Penya Playuse', '100 Media St', 01234567, 'pp@play.com', 1010);

# Internal Organisation



INSERT INTO Internal\_Organisation (Organisation\_ID, Organisation\_Name, Address, Phone\_Number, Email, Internal\_Budget\_Number, Department\_Code) VALUES

(1, 'Hi Tech', '100 Victoria Hotel', 12345678, 'vic@hotel.com', 2001, 'IT01'),

(2, 'Health Solutions', '200 Health Blvd', 23456789, 'contact@healthsolutions.com', 2002, 'HR01'),

(3, 'Tholo Energy', '300 Roma', 34567890, 'tholo@energy.com', 2003, 'ENG01'),

(4, 'Lesotho College', '400 Learn St', 45678901, 'ls@educationfirst.com', 2004, 'EDU01'),

(5, 'Standard Lesotho', '500 Money St', 56789012, 'std@bank.com', 2005, 'FIN01'),

(6, 'Travel Agency', '600 Travel Ln', 67890123, 'info@travelagency.com', 2006, 'TRV01'),

(7, 'Jays Foods', '700 Food St', 78901234, 'j@foodservices.com', 2007, 'FD01'),

(8, 'Eagle Construction Co', '800 Build St', 89012345, 'info@constructionco.com', 2008, 'CON01'),

(9, 'Retail Group', '900 Shop St', 90123456, 'e@retailgroup.com', 2009, 'RTL01'),

(10, 'Penya Playuse', '100 Media St', 01234567, 'pp@play.com', 2010, 'MD01');

# External Organisation



INSERT INTO External\_Organisation (Organisation\_ID, Organisation\_Name, Address, Phone\_Number, Email, Sponsor\_Name, External\_Budget\_Number) VALUES

(1, 'Hi Tech', '100 Victoria Hotel', 12345678, 'vic@hotel.com', 'Tech Sponsor A', 3001),

(2, 'Health Solutions', '200 Health Blvd', 23456789, 'contact@healthsolutions.com', 'Health Sponsor B', 3002),

(3, 'Tholo Energy', '300 Roma', 34567890, 'tholo@energy.com', 'Energy Sponsor C', 3003),

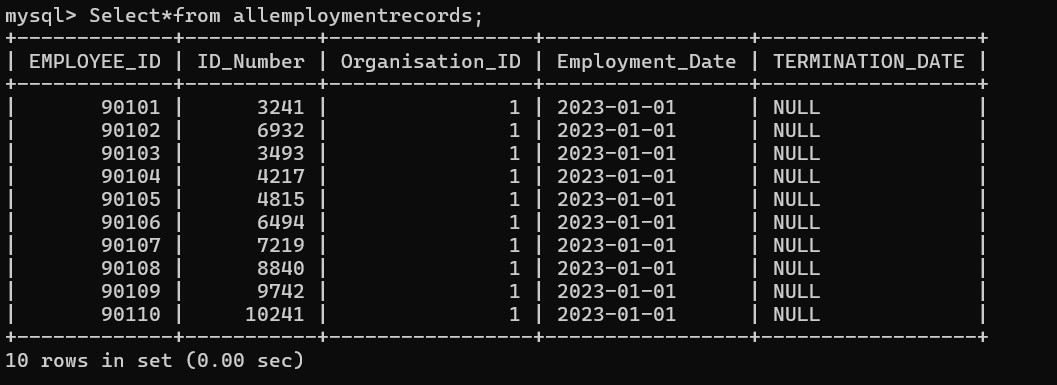
(4, 'Lesotho College', '400 Learn St', 45678901, 'ls@educationfirst.com', 'Education Sponsor D', 3004),

(5, 'Standard Lesotho', '500 Money St', 56789012, 'std@bank.com', 'Finance Sponsor E', 3005);

# Advanced SQL: Creating views

# Employment

1. All employment records



CREATE VIEW AllEmploymentRecords AS

SELECT

EMPLOYEE\_ID,

ID\_Number,

Organisation\_ID,

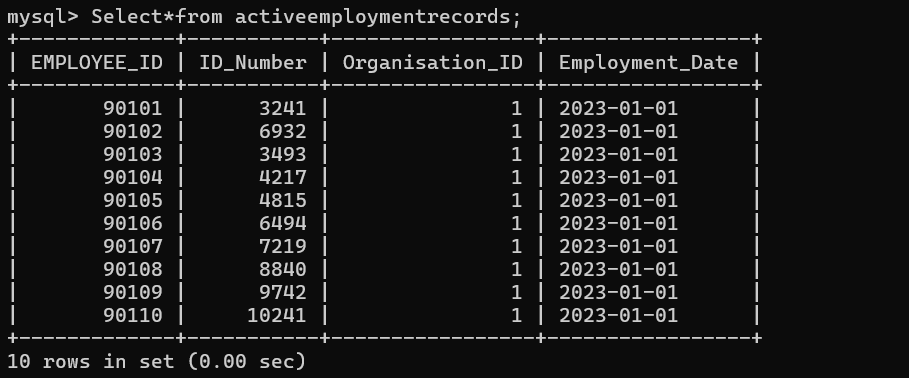
Employment\_Date,

TERMINATION\_DATE

FROM

EMPLOYMENT;

1. Active employment record



CREATE VIEW ActiveEmploymentRecords AS

SELECT

EMPLOYEE\_ID,

ID\_Number,

Organisation\_ID,

Employment\_Date

FROM

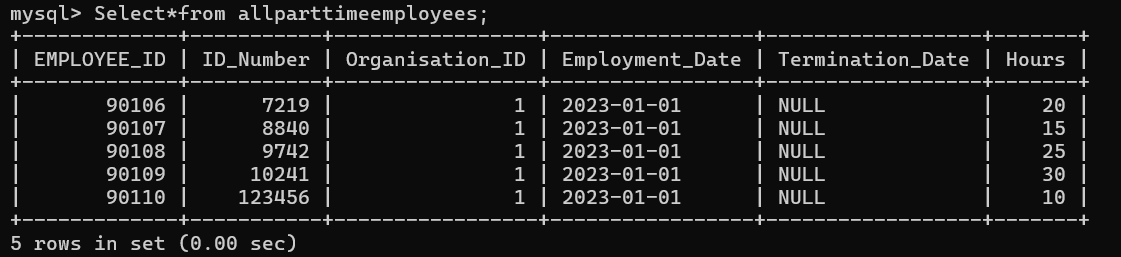
EMPLOYMENT

WHERE

TERMINATION\_DATE IS NULL;

# Part Time Employee

1. All Part Time Employees



CREATE VIEW AllPartTimeEmployees AS

SELECT

EMPLOYEE\_ID,

ID\_Number,

Organisation\_ID,

Employment\_Date,

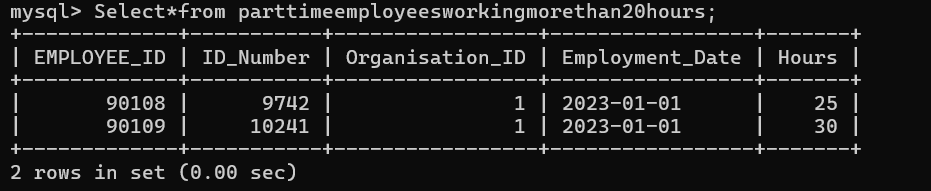
Termination\_Date,

Hours

FROM

Part\_Time\_Employee;

1. Part Time Employees Working More Than 20Hours



CREATE VIEW PartTimeEmployeesWorkingMoreThan20Hours AS

SELECT

EMPLOYEE\_ID,

ID\_Number,

Organisation\_ID,

Employment\_Date,

Hours

FROM

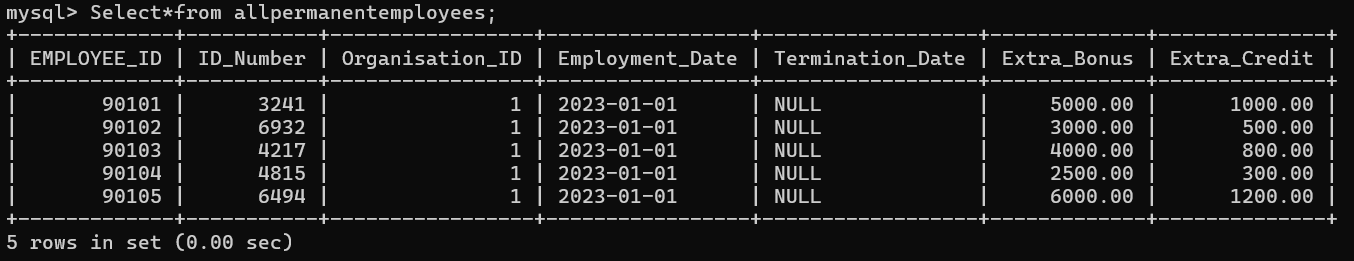
Part\_Time\_Employee

WHERE

Hours > 20;

# Permanent Employee

1. All Permanent Employees



CREATE VIEW AllPermanentEmployees AS

SELECT

EMPLOYEE\_ID,

ID\_Number,

Organisation\_ID,

Employment\_Date,

Termination\_Date,

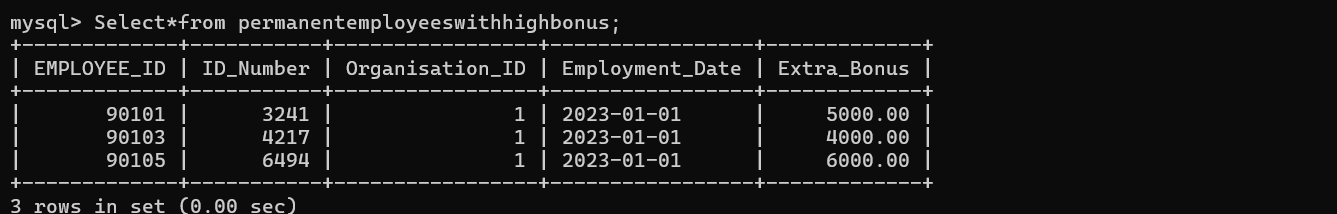
Extra\_Bonus,

Extra\_Credit

FROM

Permanent\_Employee;

1. Permanent Employees With High Bonus



CREATE VIEW PermanentEmployeesWithHighBonus AS

SELECT

EMPLOYEE\_ID,

ID\_Number,

Organisation\_ID,

Employment\_Date,

Extra\_Bonus

FROM

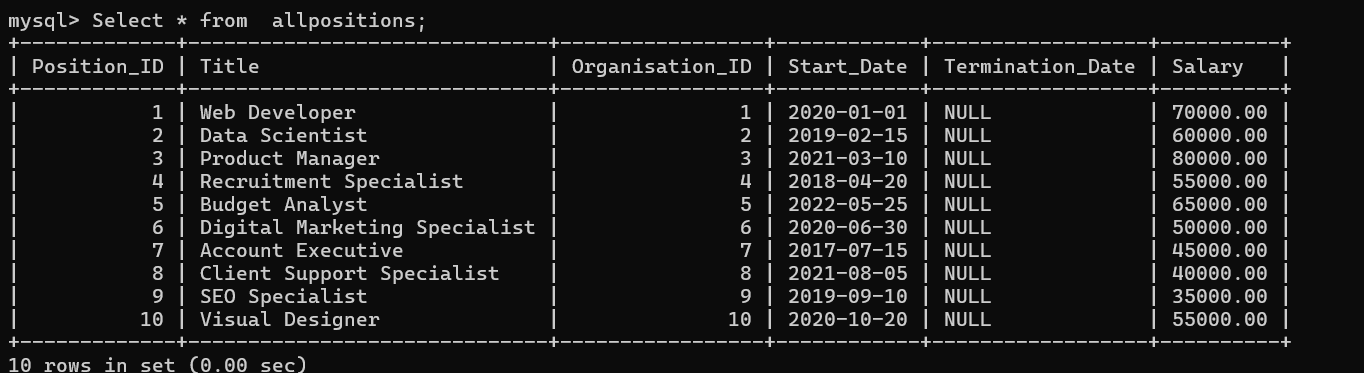
Permanent\_Employee

WHERE

Extra\_Bonus > 3000;

# Position

1. All Positions



CREATE VIEW AllPositions AS

SELECT

Position\_ID,

Title,

Organisation\_ID,

Start\_Date,

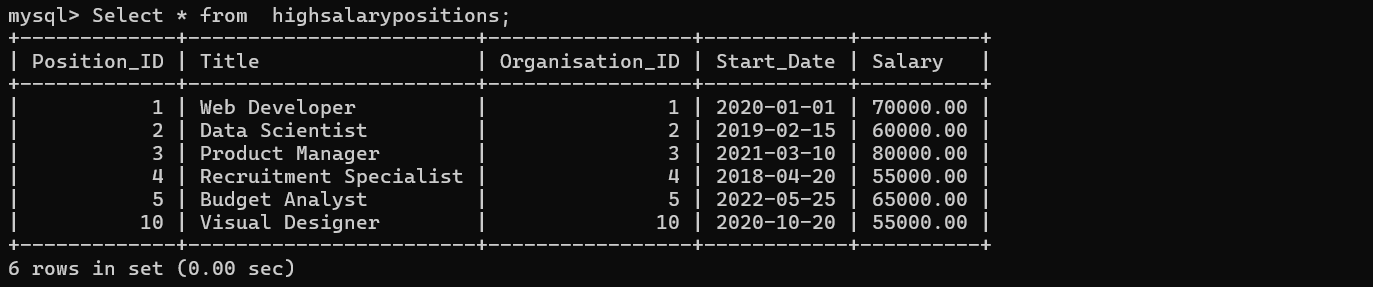
Termination\_Date,

Salary

FROM

Position;

1. High Salary Positions



CREATE VIEW HighSalaryPositions AS

SELECT

Position\_ID,

Title,

Organisation\_ID,

Start\_Date,

Salary

FROM

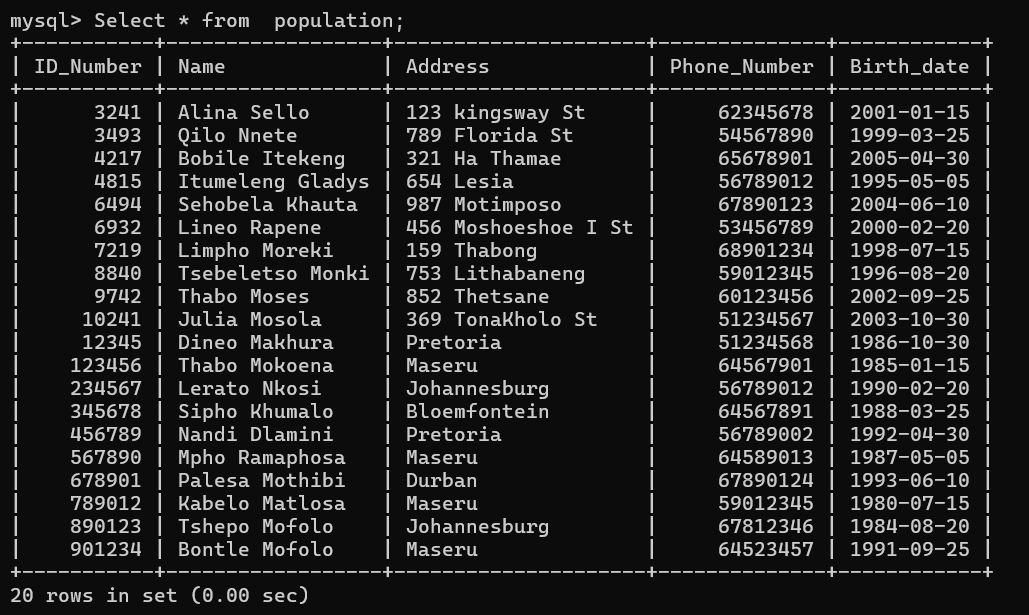
Position

WHERE

Salary > 50000;

# Person

1. Population



CREATE VIEW Population AS

SELECT

ID\_Number,

Name,

Address,

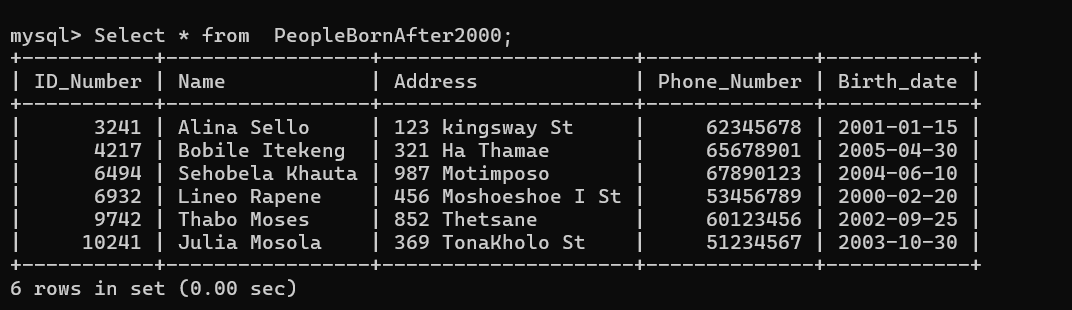
Phone\_Number,

Birth\_date

FROM

PERSON;

1. People Born After 2000



CREATE VIEW PeopleBornAfter2000 AS

SELECT

ID\_Number,

Name,

Address,

Phone\_Number,

Birth\_date

FROM

PERSON

WHERE

Birth\_date > '2000-01-01';

# Organisation

1. All Organisations



CREATE VIEW AllOrganisations AS

SELECT

Organisation\_ID,

Organisation\_Name,

Address,

Phone\_Number,

Email,

Budget\_Number

FROM

ORGANISATION;

1. Organisation With the Highest Budget Number



CREATE VIEW OrganisationsWithHighBudget AS

SELECT

Organisation\_ID,

Organisation\_Name,

Address,

Phone\_Number,

Email,

Budget\_Number

FROM

ORGANISATION

WHERE

Budget\_Number > 1000;

# Internal Organisation

1. All Internal Organisations



CREATE VIEW AllInternalOrganisations AS

SELECT

Organisation\_ID,

Organisation\_Name,

Address,

Phone\_Number,

Email,

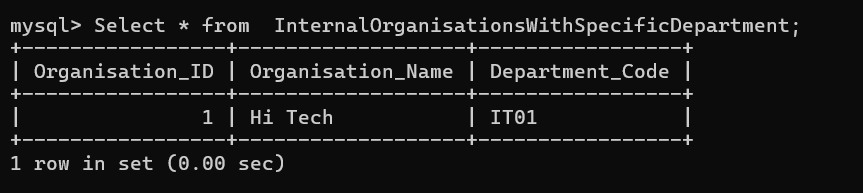
Internal\_Budget\_Number,

Department\_Code

FROM

Internal\_Organisation;

1. Internal Organisations With Specific Department



CREATE VIEW InternalOrganisationsWithSpecificDepartment AS

SELECT

Organisation\_ID,

Organisation\_Name,

Department\_Code

FROM

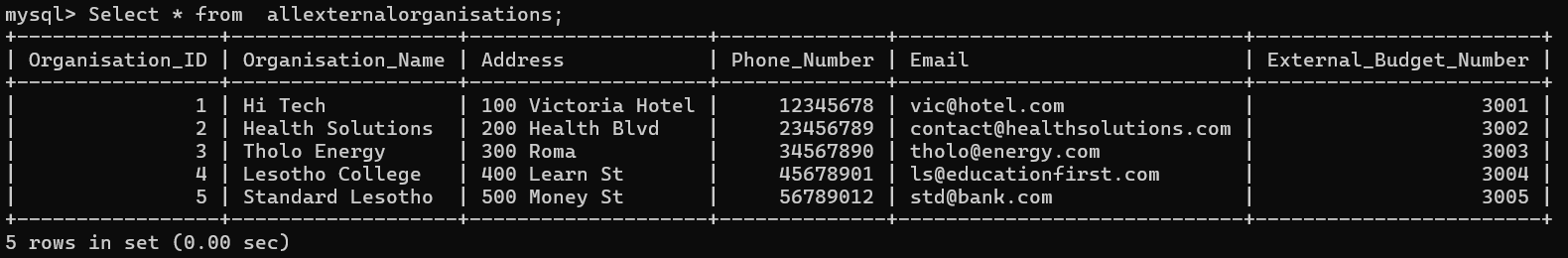
Internal\_Organisation

WHERE

Department\_Code = 'IT01';

# External Organisation

1. All External Organisations



CREATE VIEW AllExternalOrganisations AS

SELECT

Organisation\_ID,

Organisation\_Name,

Address,

Phone\_Number,

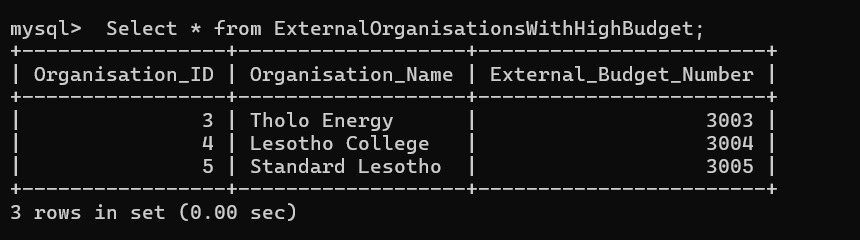
Email,

External\_Budget\_Number

FROM

External\_Organisation;

1. External Organisations With High Budget Number



CREATE VIEW ExternalOrganisationsWithHighBudget AS

SELECT

Organisation\_ID,

Organisation\_Name,

External\_Budget\_Number

FROM

External\_Organisation

WHERE

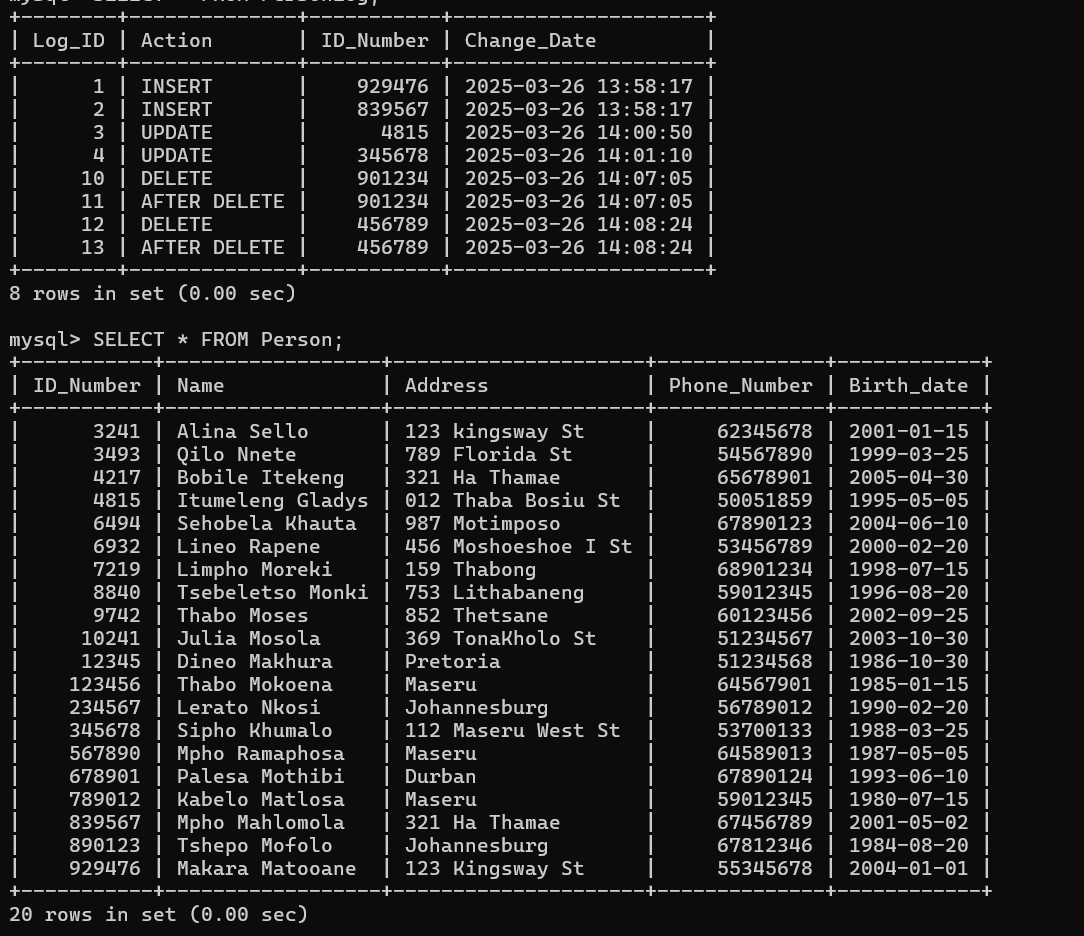
External\_Budget\_Number > 3002;

# Using Triggers

Person Table

Before





**INSERT**

INSERT INTO PERSON (ID\_Number, Name, Address, Phone\_Number, Birth\_date) VALUES

(929476, 'Makara Matooane', '123 Kingsway St', 55345678, '2004-01-01'),

(839567, 'Mpho Mahlomola', '321 Ha Thamae', 67456789, '2001-05-02');

**UPDATE**

UPDATE PERSON

SET Address = '012 Thaba Bosiu St', Phone\_Number = 50051859

WHERE ID\_Number = 4815;

UPDATE PERSON

SET Address = '112 Maseru West St', Phone\_Number = 53700133

WHERE ID\_Number = 345678;

**DELETE**

DELETE FROM PERSON

WHERE Phone\_Number = 64523457;

DELETE FROM PERSON

WHERE Phone\_Number = 56789002;

**Procedures**

1.

DELIMITER //

CREATE PROCEDURE Add\_Organisation(

IN orgID INT,

IN orgName VARCHAR(30),

IN address VARCHAR(20),

IN phoneNumber INT,

IN email VARCHAR(30),

IN budgetNumber INT

)

BEGIN

INSERT INTO ORGANISATION (Organisation\_ID, Organisation\_Name, Address, Phone\_Number, Email, Budget\_Number)

VALUES (orgID, orgName, address, phoneNumber, email, budgetNumber);

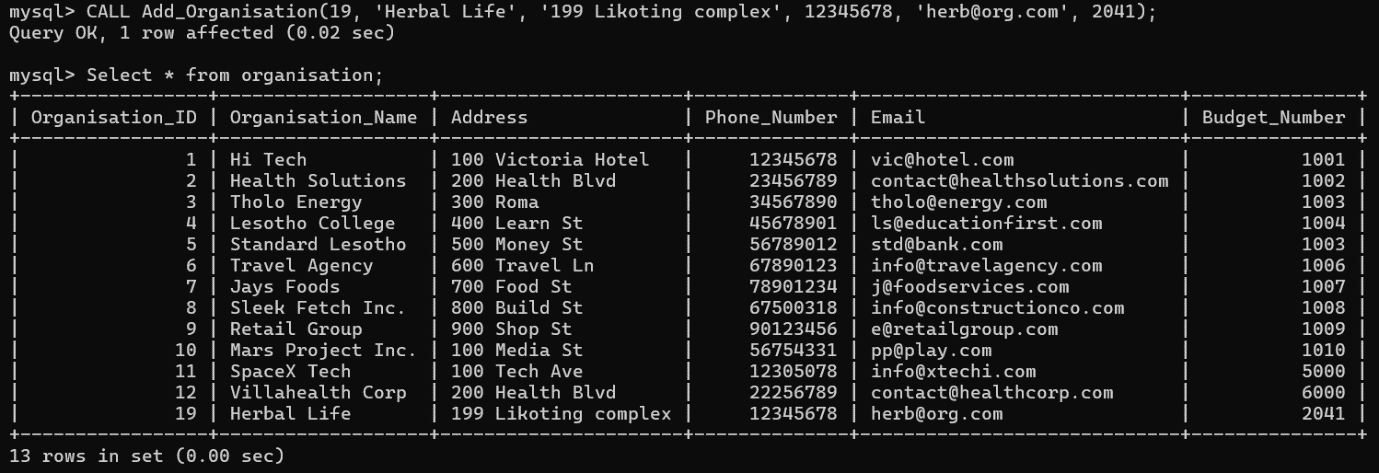
END;

//

DELIMITER ;

**TESTING THE PROCEDURE**

CALL Add\_Organisation(3, 'New Organisation', '123 New St', 12345678, 'new@organisation.com', 2000);



2.

DELIMITER //

CREATE PROCEDURE Update\_EmployeeBonus(

IN empID INT,

IN newBonus DECIMAL(10, 2)

)

BEGIN

UPDATE Permanent\_Employee

SET Extra\_Bonus = newBonus

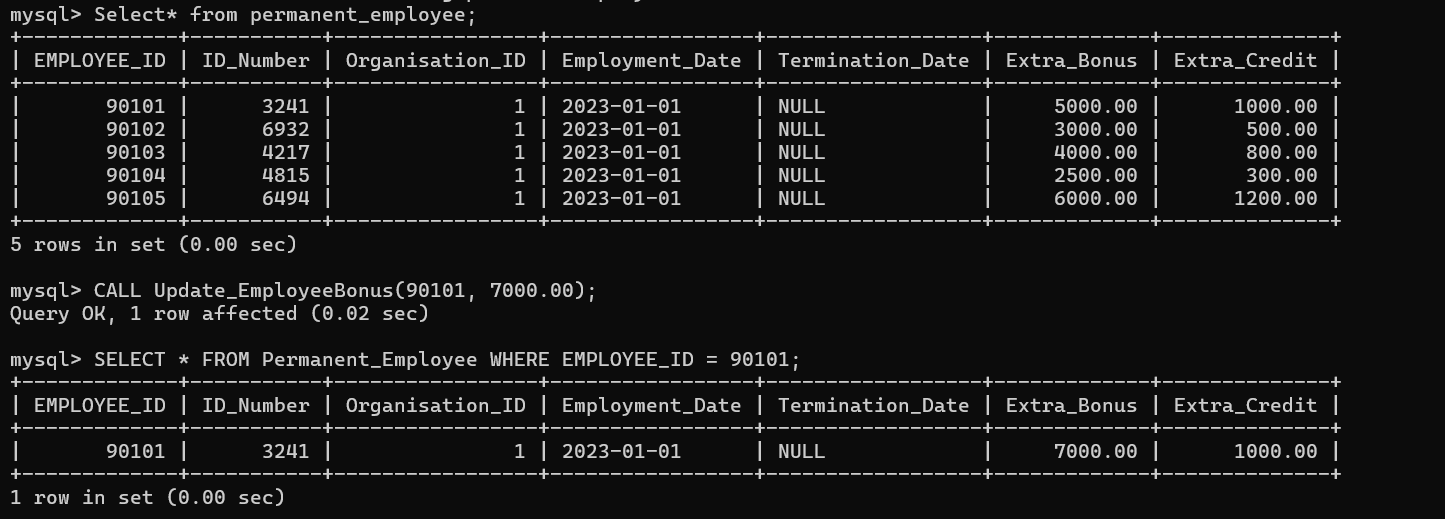
WHERE EMPLOYEE\_ID = empID;

END;

//

DELIMITER ;

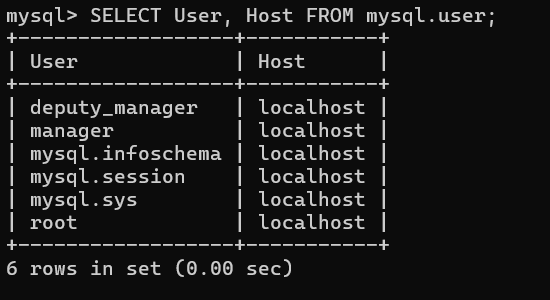
CALL Update\_EmployeeBonus(90101, 7000.00);



**User Creation and Privileges**

CREATE USER 'manager'@'localhost' IDENTIFIED BY '1111';

CREATE USER 'deputy\_manager'@'localhost' IDENTIFIED BY '1234';



**Privileges**

GRANT SELECT, INSERT ON tt\_holding.\* TO 'manager'@'localhost';

GRANT SELECT ON tt\_holding.\* TO 'deputy\_manager'@'localhost';

GRANT UPDATE ON tt\_holding.PERSON TO 'deputy\_manager'@'localhost';

