**PL/SQL**

**Que-7**

create database ex2;

use ex2;

-- Create the customers table

CREATE TABLE customers (

customer\_id int PRIMARY KEY,

name VARCHAR(255),

address VARCHAR(255)

);

-- Create the customer\_balances table

CREATE TABLE customer\_balances (

customer\_id int PRIMARY KEY,

balance int

);

-- Create the employees table

CREATE TABLE employees (

employee\_id int PRIMARY KEY,

name VARCHAR(255),

position VARCHAR(255),

salary int

);

-- Create the accounts table

CREATE TABLE accounts (

account\_id int PRIMARY KEY,

customer\_id int,

balance int,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

DELIMITER $$

CREATE PROCEDURE AddCustomer(

IN p\_customer\_id INT,

IN p\_name VARCHAR(255),

IN p\_address VARCHAR(255)

)

BEGIN

INSERT INTO customers (customer\_id, name, address) VALUES (p\_customer\_id, p\_name, p\_address);

END $$

DELIMITER ;

DROP TABLE IF EXISTS accounts;

-- Create the customers table

CREATE TABLE IF NOT EXISTS customers (

customer\_id INT PRIMARY KEY,

name VARCHAR(255),

address VARCHAR(255)

);

-- Create the customer\_balances table

CREATE TABLE IF NOT EXISTS customer\_balances (

customer\_id INT PRIMARY KEY,

balance DECIMAL(10, 2),

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

-- Create the employees table

CREATE TABLE IF NOT EXISTS employees (

employee\_id INT PRIMARY KEY,

name VARCHAR(255),

position VARCHAR(255),

salary DECIMAL(10, 2)

);

-- Create the accounts table

CREATE TABLE IF NOT EXISTS accounts (

account\_id INT PRIMARY KEY,

customer\_id INT,

balance DECIMAL(10, 2),

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

DELIMITER $$

CREATE PROCEDURE AddCustomer(

IN p\_customer\_id INT,

IN p\_name VARCHAR(255),

IN p\_address VARCHAR(255)

)

BEGIN

INSERT INTO customers (customer\_id, name, address) VALUES (p\_customer\_id, p\_name, p\_address);

END $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE UpdateCustomer(

IN p\_customer\_id INT,

IN p\_name VARCHAR(255),

IN p\_address VARCHAR(255)

)

BEGIN

UPDATE customers

SET name = p\_name, address = p\_address

WHERE customer\_id = p\_customer\_id;

END $$

DELIMITER ;

DELIMITER $$

CREATE FUNCTION GetCustomerBalance(p\_customer\_id INT)

RETURNS DECIMAL(10,2)

DETERMINISTIC

BEGIN

DECLARE v\_balance DECIMAL(10,2);

SELECT balance INTO v\_balance

FROM customer\_balances

WHERE customer\_id = p\_customer\_id;

RETURN v\_balance;

END $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE HireEmployee(

IN p\_employee\_id INT,

IN p\_name VARCHAR(255),

IN p\_position VARCHAR(255),

IN p\_salary DECIMAL(10,2)

)

BEGIN

INSERT INTO employees (employee\_id, name, position, salary)

VALUES (p\_employee\_id, p\_name, p\_position, p\_salary);

END $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE UpdateEmployee(

IN p\_employee\_id INT,

IN p\_name VARCHAR(255),

IN p\_position VARCHAR(255),

IN p\_salary DECIMAL(10,2)

)

BEGIN

UPDATE employees

SET name = p\_name, position = p\_position, salary = p\_salary

WHERE employee\_id = p\_employee\_id;

END $$

DELIMITER ;

DELIMITER $$

CREATE FUNCTION CalculateAnnualSalary(p\_employee\_id INT)

RETURNS DECIMAL(10,2)

DETERMINISTIC

BEGIN

DECLARE v\_salary DECIMAL(10,2);

SELECT salary \* 12 INTO v\_salary

FROM employees

WHERE employee\_id = p\_employee\_id;

RETURN v\_salary;

END $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE OpenAccount(

IN p\_account\_id INT,

IN p\_customer\_id INT,

IN p\_initial\_balance DECIMAL(10,2)

)

BEGIN

INSERT INTO accounts (account\_id, customer\_id, balance)

VALUES (p\_account\_id, p\_customer\_id, p\_initial\_balance);

END $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE CloseAccount(

IN p\_account\_id INT

)

BEGIN

DELETE FROM accounts

WHERE account\_id = p\_account\_id;

END $$

DELIMITER ;

DELIMITER $$

CREATE FUNCTION GetTotalBalance(p\_customer\_id INT)

RETURNS DECIMAL(10,2)

DETERMINISTIC

BEGIN

DECLARE v\_total\_balance DECIMAL(10,2);

SELECT SUM(balance) INTO v\_total\_balance

FROM accounts

WHERE customer\_id = p\_customer\_id;

RETURN v\_total\_balance;

END $$

DELIMITER ;

CALL AddCustomer(1, 'John Doe', '123 Elm Street');

SELECT GetCustomerBalance(1);

CALL HireEmployee(1, 'Jane Smith', 'Manager', 5000);

SELECT CalculateAnnualSalary(1);

CALL OpenAccount(1, 1, 1000.00);

SELECT GetTotalBalance(1);