Banking Management System using Oracle SQL & PL/SQL

Objective:

To design and implement a simple core banking system that manages:

Customers

Bank accounts

Deposits & withdrawals

Transaction history

Balance validation

This project demonstrates real-world database design, PL/SQL programming, and transaction control.

Technologies used :

Database: Oracle SQL

Language: PL/SQL

Concepts Used:

Tables & Constraints

Sequences

Stored Procedures

Functions

Triggers

Exception Handling

Transactions (COMMIT / ROLLBACK)

Database Design:

CUSTOMERSS TABLE

Stores customer personal details.

customer_id (Primary Key)

name

city

contact

ACCOUNTS TABLE

Stores bank account details for customers.

account_id (Primary Key)

customer_id (Foreign Key)

account_type

balance

TRANSACTIONS TABLE

Stores all deposit and withdrawal history.

txn_id (Primary Key)

account_id (Foreign Key)

txn_type (DEPOSIT / WITHDRAW)

amount

txn_date

SEQUENCES USED

Sequence Name Purpose

seq_customer Auto-generate customer_id

seq_account Auto-generate account_id

seq_txn Auto-generate transaction_id

Core Functionalities:

CREATE CUSTOMER

Inserts new customer data

Automatically generates customer ID

Procedure: create_customer

OPEN ACCOUNT

Creates a new bank account for a customer

Uses opening balance

Procedure: open_account

DEPOSIT MONEY

Adds amount to account balance

Automatically logs transaction

Validates account existence

Procedure: deposit_money

WITHDRAW MONEY

Deducts amount from balance

Prevents overdraft

Automatically logs transaction

Procedure: withdraw_money

TRANSACTION LOGGING

Every deposit or withdrawal automatically inserts a record into transactions table

Uses sequence for transaction ID

Procedure: add_transaction

Balance Validation Trigger

Prevents negative balance updates

Stops illegal withdrawals

Trigger: check_balance

Balance Enquiry

Returns current balance of an account

Function: get_balance

Execution Flow (Real Banking Flow)

Create Customer

Open Account

Deposit Money

Balance updated

Transaction recorded

Withdraw Money

Balance checked

Trigger prevents negative balance

Transaction recorded

Check Balance

View Transaction History

Key Features & Highlights

- Real-world banking logic
- Fully automated transaction logging
- Strong data integrity using constraints
- Exception handling for errors
- Trigger-based balance protection