**PART B – Micro–Project Report**

**Title of Micro-Project:** Bank Management System In DBMS.

**1.0 Rationale**: Bank Management System is based on the concept of recording customers’ account details. Here the user can perform all the tasks like creating an account, depositing the amount, withdrawing the amount, checking the balance, viewing all account holders’ detail, closing an account, and modifying an account. There's no login system for this project

**2.0 Course Outcomes Addressed:**

* + Design a ER diagram on Bank management system.
  + Create and Manage the Database using SQL command.

**3.0 Literature Review:**

1. ”Database systems, concepts, design and Applications” by S.K.Singh, person education.
2. “Database management systems” by Raghu Ramakrishna, Johannes Gehrke, McGraw Hill
3. “Fundamentals of Database Systems” by Elmsari, Navathe, 5th Edition, Pearson education(2008).

**4.0 Actual Methodology Used:**

-- Program to create a bank management system

DECLARE -- Declare variables

account\_name VARCHAR2(20);

account\_no NUMBER;

balance NUMBER;

deposit NUMBER;

withdraw NUMBER;

-- Create table CREATE TABLE bank\_accounts ( -- Define columns account\_name VARCHAR2(20), account\_no NUMBER, balance NUMBER ); --

Start procedure

PROCEDURE create\_account ( -- Define parameters p\_account\_name VARCHAR2, p\_account\_no NUMBER, p\_balance NUMBER )

AS -- Insert record INSERT INTO bank\_accounts VALUES (p\_account\_name, p\_account\_no, p\_balance);

BEGIN -- Ask user for input DBMS\_OUTPUT.PUT\_LINE('Enter account name: '); account\_name := &account\_name; DBMS\_OUTPUT.PUT\_LINE('Enter account number: '); account\_no := &account\_no; DBMS\_OUTPUT.PUT\_LINE('Enter initial balance: ');

balance := &balance; -- Create account create\_account (account\_name, account\_no, balance); DBMS\_OUTPUT.PUT\_LINE('Account created successfully!');

END;

PROCEDURE deposit\_money ( p\_account\_no NUMBER, p\_deposit NUMBER ) AS -- Update balance UPDATE bank\_accounts SET balance = balance + p\_deposit WHERE account\_no = p\_account\_no; BEGIN -- Ask user for input DBMS\_OUTPUT.PUT\_LINE('Enter account number: ');

account\_no := &account\_no; DBMS\_OUTPUT.PUT\_LINE('Enter amount to deposit: ');

deposit := &deposit; -- Deposit money deposit\_money (account\_no, deposit); DBMS\_OUTPUT.PUT\_LINE('Money deposited successfully!');

END;

PROCEDURE withdraw\_money ( p\_account\_no NUMBER, p\_withdraw NUMBER )

AS -- Update balance UPDATE bank\_accounts

SET balance = balance - p\_withdraw

WHERE account\_no = p\_account\_no; BEGIN -- Ask user for input DBMS\_OUTPUT.PUT\_LINE('Enter account number: ');

account\_no := &account\_no; DBMS\_OUTPUT.PUT\_LINE('Enter amount to withdraw: '); withdraw := &withdraw; -- Withdraw money withdraw\_money (account\_no, withdraw); DBMS\_OUTPUT.PUT\_LINE('Money withdrawn successfully!');

END;

BEGIN -- Call procedures create\_account; deposit\_money; withdraw\_money; -- Display table DBMS\_OUTPUT.PUT\_LINE('Accounts: ');

SELECT \* FROM bank\_accounts;

END;

/

**5.0 Actual Resources Used:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr No** | **Name of Resource /Material** | **Specifications** | **Qty** | **Remarks** |
| **1.** | Microsoft Edge | Browser | 1 | - |
| 2. | Oracle | Specifications | 1 | **-** |
| 3. | Laptop | Intel corei5,1Tb SSD,8GB RAM | 1 | **-** |

**6.0 Output of Micro-project:**

Enter account name: John Smith

Enter account number: 123456

Enter initial balance: 1000

Account created successfully!

**select \* from bank\_account;**

|  |  |  |
| --- | --- | --- |
| **Account Name** | **Account Number** | **Initial Balance** |
| John Smith | 123456 | 1000 |

Enter account number: 123456

Enter amount to deposit: 500

Money deposited successfully!

**select \* from bank\_account;**

|  |  |  |
| --- | --- | --- |
| **Account name** | **Account number** | **Initial balance** |
| John smith | 123456 | 1500 |

Enter account number: 123456

Enter amount to withdraw: 200

Money withdrawn successfully!

**select \* from bank\_account;**

|  |  |  |
| --- | --- | --- |
| **Account Name** | **Account Number** | **Initial Balance** |
| John Smith | 123456 | 1200 |

**7.0 Skills Developed/learning out of this Micro–Project:**

* We developed programming skills.
* We developed searching skills.
* We developed our teamwork skills.

**8.0 Application of this Micro-Project :**

* Creating the bank account for the customer.
* Deposit amount for the customer.
* Withdraw the amount for the customer.
* Balance Enquiry by the customer.
* Show the account holder's full detail.
* Closing or terminating a bank account.