Assessment :

Pseudo code 10 Marks

Q1 : Write a Pseudo code for verify if number entered is prime number.

BEGIN

DECLARE ‘Variable’, NUM as integer

PROMPT ‘Number’

STORE ‘Number’ in NUM variable

IF NUM is divisible by 1 AND NUM divisible by NUM

THEN Display “Number”, NUM “is a prime number”

ELSE

DISPLAY “Number”, NUM “is not a prime number”

END-IF

END

Q2: Write a Pseudo code for transferring amount from one account to another. [Should validate whether both accounts exists]

BEGIN

DECLARE ‘Variable’ ACCT\_FROM, ACCT\_TO, AMT\_FROM as integer

PROMPT ‘Get Account number of Transferor’ and STORE in ACCT\_FROM variable

PROMPT ‘GET Account number of Transferee’ and STORE in ACCT\_TO variable

PROMPT ‘GET Amount to be transferred’ and STORE in AMT\_FROM variable

--Validate user credentials of Transferor account—

IF LOGIN ID AND PASSWORD of ACCT\_FROM = ACTIVE

--Check if Transferor account is in ACTIVE status—

IF ACCT\_FROM = ACTIVE

--Check if amount is within transferable limit--

IF AMT\_FROM <= Maximum amount that can be transferred AND AMT\_FROM <= Available Balance

THEN Transfer the amount in ACCT\_TO account

ELSE

DISPLAY “Amount that is being transferred exceeds the maximum allowed limit OR Insufficient Balance”

END-IF

ELSE

DISPLAY “Transferor account number is does not exist. Please contact bank customer care.”

END-IF

ELSE

DISPLAY “User ID and/or Password is invalid. Please try again OR contact customer care for help”

END

RDBMS: 30 Marks

Q1. Write SQL Query to create following tables [DO NOT CREATE PRIMARY / FOREIGN KEYS ]

Customer: CustomerId, Fullname, address, city, pan number

Account: accountNo, accType, balance, customerId

--SQL to create ‘Customer’ table--

CREATE TABLE CUSTOMER (

CUSTOMERID VARCHAR (5),

FULLNAME VARCHAR (50),

ADDRESS VARCHAR (100),

CITY VARCHAR (10),

PAN\_NUMBER VARCHAR (20)

);

--SQL to create ‘Account’ table--

CREATE TABLE ACCOUNT (

ACCOUNTNO VARCHAR (10),

ACCTTYPE VARCHAR (10),

BALANCE NUMBER (10,2),

CUSTOMERID VARCHAR (5)

);

Q2. Write SQL to insert following records in Customer & Account tables:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C1002 | Rajiv Bhatia | Xyz Path, Chandni chowk | Delhi | AXNSS 1234 A |
| C1003 | Alia Bhatt | Khar | Mumbai | SZAXS 5656 B |
| C1004 | Vijay Deol | Bandra | Mumbai | APOI 5675 A |
| C1005 | Ajay Deol | Bandra | Mumbai | AUIO 7676 K |

--SQL to insert record in Customer table –

INSERT INTO CUSTOMER (CUSTOMERID, FULLNAME, ADDRESS, CITY, PAN\_NUMBER)

VALUES ('C1002', 'Rajiv Bhatia', 'Xyz Path', 'Delhi', 'AXNSS 1234 A');

INSERT INTO CUSTOMER (CUSTOMERID, FULLNAME, ADDRESS, CITY, PAN\_NUMBER)

VALUES ('C1003', 'Alia Bhatt', 'Khar', 'Mumbai', 'SZAXS 5656 B');

INSERT INTO CUSTOMER (CUSTOMERID, FULLNAME, ADDRESS, CITY, PAN\_NUMBER)

VALUES ('C1004', 'Vijay Deol', 'Bandra', 'Mumbai', 'APOI 5675 A');

INSERT INTO CUSTOMER (CUSTOMERID, FULLNAME, ADDRESS, CITY, PAN\_NUMBER)

VALUES ('C1005', 'Ajay Deol', 'Bandra', 'Mumbai', ' AUIO 7676 K ');

|  |  |  |  |
| --- | --- | --- | --- |
| SB122666 | Savings | 67000 | C1002 |
| CB565556 | Current | 786928.98 | C1002 |
| SB876565 | Savings | 547899.90 | C1004 |
| SB565722 | Savings | 67600 | C1003 |
| SB757676 | Savings | 66197.88 | C1003 |
| SB166778 | Current | 16000 | C1008 |

--SQL to insert record in Account table –

INSERT INTO ACCOUNT (ACCOUNTNO, ACCTTYPE, BALANCE, CUSTOMERID)

VALUES ('SB122666', ‘Savings', 67000, 'C1002’);

INSERT INTO ACCOUNT (ACCOUNTNO, ACCTTYPE, BALANCE, CUSTOMERID)

VALUES ('CB565556', ‘Current', 786928.98, 'C1002’);

INSERT INTO ACCOUNT (ACCOUNTNO, ACCTTYPE, BALANCE, CUSTOMERID)

VALUES ('SB876565', ‘Savings’, 547899.90, 'C1004’);

INSERT INTO ACCOUNT (ACCOUNTNO, ACCTTYPE, BALANCE, CUSTOMERID)

VALUES ('SB565722', ‘Savings’, 67000, 'C1003’);

INSERT INTO ACCOUNT (ACCOUNTNO, ACCTTYPE, BALANCE, CUSTOMERID)

VALUES ('SB757676', ‘Savings’, 66197.88, 'C1003’);

INSERT INTO ACCOUNT (ACCOUNTNO, ACCTTYPE, BALANCE, CUSTOMERID)

VALUES ('SB166778', ‘Current’, 16000, 'C1008’);

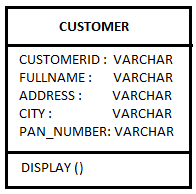
Q3. Write a Left Join to get all customers and accounts. Join should display all customers [Even those who DO NOT have any account].

SELECT CUSTOMERID, FULLNAME, ACCOUNTNO, ACCTTYPE, BALANCE FROM CUSTOMER LEFT JOIN ACCOUNT USING (CUSTOMERID)

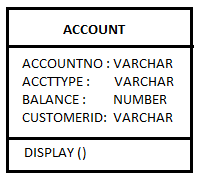
UML 10 Marks

Q1. For RDBMS Question #1, Create Class Diagram for Both tables.

--Class diagram for CUSTOMER table—



--Class diagram for ACCOUNT table—



Q2. Create an Activity Diagram to explain fund transfer.

HINT: Fund transfer is possible from Any account type to any other account type.

Must validate existence of both account

Must validate account balance before transfer

Must update balance after transaction completes.

--Activity Diagram –

Verification Failed

Display “UserID and/or Password not correct. Please enter again or try later”

Prompt and get ‘UserID’ and ‘Password’

Verify the ‘UserID’ and Password’ entered

S

Verification passed

Prompt and get user transferor account number

Send Confirmation to both transferor and transferee

Verify transferee account

Outside transfer limit

S

S

Display “Account number is invalid”. Please enter again or try later”

NO

S

Update the balance

Transfer Amount

Account verified

Prompt and get user transferee account number

Yes

Display “Insufficient Balance”.

No

Within transferable limit

If available balance >=Amount being transferred

Display “Please re-enter amount as it is not within transferable limit”.

Display “Account number is invalid”. Please enter again or try later”

Verify if amount is within the allowed transferable limit

Prompt and get amount to be transferred

Verification passed

Verification Failed

Verify Account Number from DB