Pseudo code 10 Marks

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

Begin

Declare UserInput as int

Get input from user

Check if the entered number is divisible by 1

Check if the entered number is divisible by itself

If No – Display msg “Entered number is not a Prime Number”

If Yes – Display msg “Entered number is a Prime Number”

End

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

Begin

Declare Amount, AccountNo as int, IFSCcode as Varchar, Branch as Varchar

Validate if the transferor’s account exists and if Users online account is Active

Validate if the transferee’s account exists

Get input from the user for the account (A/c No.) in which the amount has to be transferred

Get input from the user for IFSC code and branch location

Add beneficiary to transferor’s account

Validate if transferee’s account is valid and active

If No, Display error “Account Not Active or Invalid”

If Yes

Get input from the user for the amount which has to be transferred

Check sufficient balance availability in transferor’s account

If No, Display Error “Insufficient funds in account for this transfer”

If Yes, Display message “Funds Successfully Transferred”

Update the transferor’s account balance by reducing the amount with the amount that was transferred.

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

* Create table Customer

(CustomerId int (10),

Fullname Varchar (30),

address Varchar (50),

city Varchar(10),

pan number int(10));

* Create table Account

(accountNo int (15),

acctType Varchar (30),

balance int (100),

customerid int(10));

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 |

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

* INSERT INTO Customer (CustomerId, Fullname, address, city, pan number)

Values (‘C1002’, ‘Rajiv Bhatia’, ‘Xyz Path, Chandni chowk’, ‘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, ‘Ajjay Deol’, ‘Bandra’, ‘Mumbai’, ‘AUIO 7676 K’);

* 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’, 786928.98, ‘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].

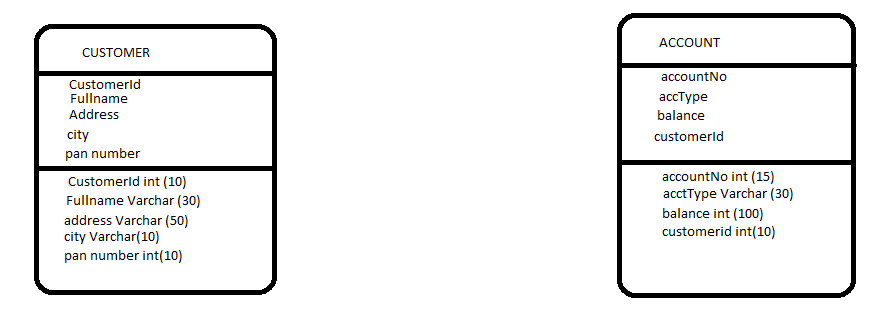
SELECT \* FROM CUSTOMER LEFT JOIN ACCOUNT USING (customerid) JOIN LOCATIONS USING (LOCATION\_ID);

SELECT \* FROM customer Left Join account ON customer.customerid = account.customerid;

=============================================================================

UML 10 Marks

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



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.

