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

Q 1. PSEUDO CODE TO VERIFY GIVNE NUMBER IS PRIME OR NOT

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

BEGIN

DECLARE i, num AS INTEGER

DECLARE flag AS BOOLEAN

PROMPT "ENTER THE NUMBER" AND STORE IN num

SET BOOLEAN flag = false;

FOR ( i = 2 ; i = num/2; i++)

IF(num mod i == 0) THEN

flag = true;

END IF

END FOR

IF(!flag) THEN

DISPLAY "THE NUMBER IS PRIME NUMBER"

ELSE

DISPLAY "THE IS NOT PRIME NUMBER"

END IF

END

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

Q 2. PSEUDO CODE FOR TRANSFERRING AMOUNT FROM ONE ACCOUNT TO ANOTHER

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

BEGIN

DECLARE acc1, acc2, amt AS INTEGER

DECLARE name, ifsccode AS STRING

DECLARE flag as BOOLEAN

PROMPT "ENTER THE PAYER ACCOUNT NUMBER" AND STORE IN acc1

PROMPT "ENTER THE PAYEE ACCOUNT NUMBER" AND STORE IN acc2

--CHECK FOR ACCOUNT EXIST

IF (acc1 IS ACTIVE) THEN

DISPLAY "WELCOME"

IF (acc2 IS EXIST) THEN

DISPLAY "PAYEE ACCOUNT EXIST!."

ELSE

DISPLAY "PAYEE ACCOUNT DOES NOT EXIST!."

END IF

ELSE

DISPLAY "PAYER ACCOUNT DOES NOT EXIST!."

END IF

--CHECK FOR TRANSFER AMOUNT

PROMPT "ENTER THE PAYEE NAME AND IFSC CODE" AND STORE IN name, ifsccode

IF (name IS EXIST) THEN

IF (ifsccode IS EXIST) THEN

EXIT

ELSE

DISPLAY "IFSC CODE DOES NOT EXIST"

END IF

ELSE

DISPLAY "PAYEE NAME DOES NOT EXIST"

END IF

-- FUND TRANSFER

PROMPT "ENTER THE AMOUNT" AND STORE IN amt

IF (AMOUNT IS NON NUMERIC)

DISPLAY "PLEASE ENTER CORRECT AMOUNT"

ELSE IF (AMOUNT AVAILABILITY IS TRUE) THEN

FUND TRANSFER DONE.

ELSE

DISPLAY "AMOUNT IS NOT SUFFIENCIENT!"

END IF

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

RDBMS

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

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

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

Customer:

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

CREATE TABLE CUSTOMERS (

CustomerId VARCHAR(5) NOT NULL,

FullName VARCHAR(30) NOT NULL,

Address VARCHAR(30) NOT NULL,

City VARCHAR(10) NOT NULL,

PANNUMBER VARCHAR(15) NOT NULL

);

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

Account :

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

CREATE TABLE ACCOUNTS (

accountNo VARCHAR(12) NOT NULL,

accType CHAR(10) NOT NULL,

balance NUMBER(20),

CustomerId VARCHAR(5) NOT NULL

);

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

INSERT FOR CUSTOMERS

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

Insert into CUSTOMERS(CustomerId,FullName,Address,City,PANNUMBER)

values ('C1002','Rajiv Bhatia','Xyz Path, Chandni chowk','Delhi','AXNSS 1234 A');

Insert into CUSTOMERS(CustomerId,FullName,Address,City,PANNUMBER)

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

Insert into CUSTOMERS(CustomerId,FullName,Address,City,PANNUMBER)

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

Insert into CUSTOMERS(CustomerId,FullName,Address,City,PANNUMBER)

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

=====================================================================================INSERT FOR ACCOUNTS

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

Insert into ACCOUNTS(ACCOUNTNO,ACCTYPE,BALANCE,CUSTOMERID)

values ('SB122666','Savings','67000','C1002');

Insert into ACCOUNTS(ACCOUNTNO,ACCTYPE,BALANCE,CUSTOMERID)

values ('CB565556','Current','786928.98','C1002');

Insert into ACCOUNTS(ACCOUNTNO,ACCTYPE,BALANCE,CUSTOMERID)

values ('SB876565','Savings','547899.90','C1004');

Insert into ACCOUNTS(ACCOUNTNO,ACCTYPE,BALANCE,CUSTOMERID)

values ('SB565722','Savings','67600','C1003');

Insert into ACCOUNTS(ACCOUNTNO,ACCTYPE,BALANCE,CUSTOMERID)

values ('SB757676','Savings','66197.88','C1003');

Insert into ACCOUNTS(ACCOUNTNO,ACCTYPE,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].

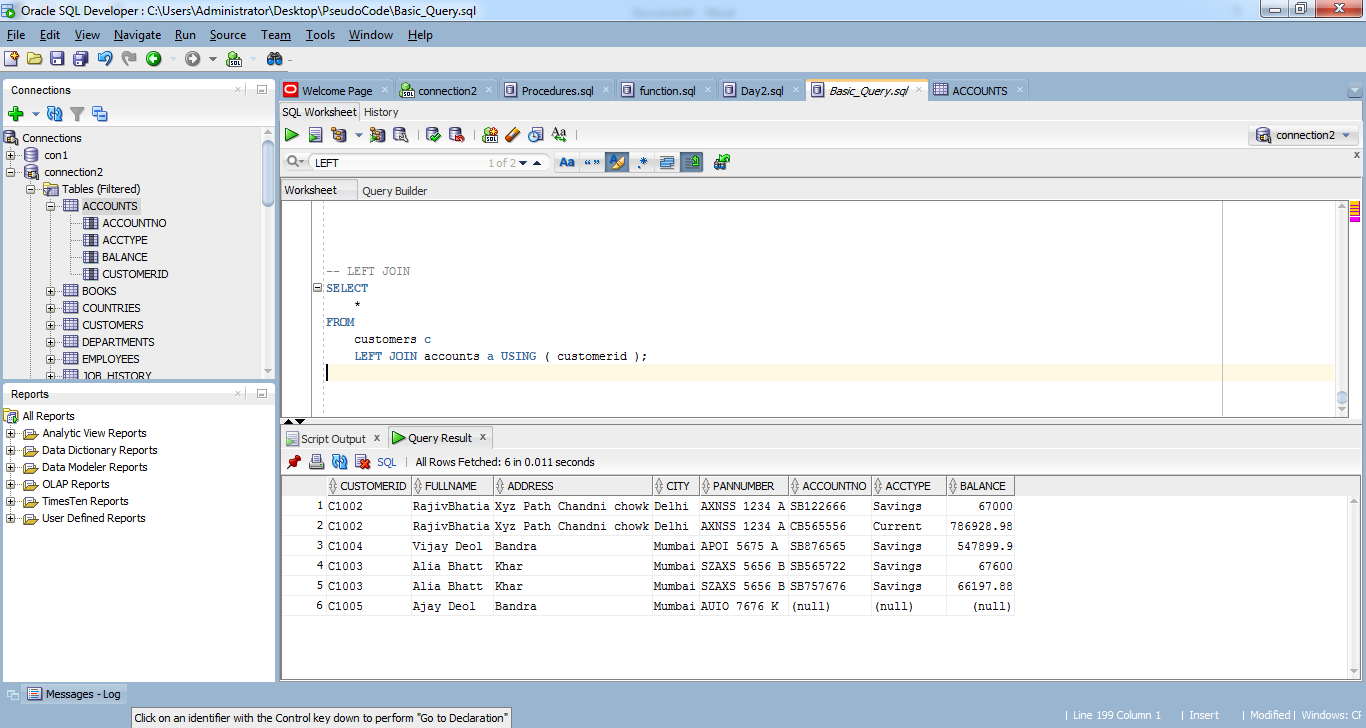
ANS:

**SELECT**

\*

**FROM**

customers c **LEFT JOIN** accounts a USING ( customerid );



**UML**

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

|  |
| --- |
| **Customer** |
| **Customer ID : Varchar,**  **Full name : Char,**  **Address : Varchar,**  **City: Varchar,**  **pan number: Varchar** |
| **SQL Query**  **1.Create()**  **2. Insert()**  **3. Select() to fetch all the records using Left Join** |

|  |
| --- |
| **Account** |
| **accountNo**: Number, **accType**: Varchar, **balance**: Float, **customerId :** Varchar |
| **SQL Query**  **1.Create()**  **2. Insert()**  **3. Select() to fetch all the records using Left Join** |

Q2. Create an Activity Diagram to explain fund transfer.

Yes

No

Invalid Payee Details

Is Payee Account Valid

Enter Payee Account Number

Invalid Accounts

No

Yes

Is

Is payer account is Valid

**Enter PAYER ACCOUNT NUMBER**

No

Check Funds

Not a sufficient funds

No

Yes

Transfer funds

Yes

End of the flow

Display Transaction Successful

Transaction failed

No

Check Transactions