DDL (Data Definition Language)

Nucleus Software wants to create a Lending Management System for the banks to take care of Loan needs of the customer

1. Create the table **Customer** based on the following instance chart.
   Name the table: **LMS_CUSTOMER_M**

Name	Null?	Type
Customer_ID	NOT NULL	VARCHAR(20)
First_Name	NOT NULL	VARCHAR(20)
Last_Name		VARCHAR(20)
Gender	NOT NULL	VARCHAR(1)
Date_Of_Birth	NOT NULL	DATE
Contact_Number	NOT NULL	VARCHAR(10)
Email_Address	NOT NULL	VARCHAR(40)
Monthly_Income	NOT NULL	NUMBER(8,2)
Profession		VARCHAR(40)
Total_Monthly_Expense	NOT NULL	NUMBER(8,2)
Designation		VARCHAR(40)
Company_Name		VARCHAR(40)

2. Create the table **LoanProductCategory** based on the following instance chart.
   Name the table: **LMS_PRODUCT_CATEGORY_M**

Name	Null?	Type	Description
Category_ID	NOT NULL	VARCHAR(10)	
Category_Name	NOT NULL	VARCHAR(50)	
Asset_Based	NOT NULL	VARCHAR (1)	Value – Y/N
Description		VARCHAR(150)	
SecuredLoan	NOT NULL	VARCHAR(2)	S – Secured US - Unsecured

3. Create the table **LoanProduct** based on the following instance chart.
   Name the table: **LMS_Product_M**

Name	Null?	Type
Product_Code	NOT NULL	VARCHAR(20)
Product_Name	NOT NULL	VARCHAR(100)
Product_Description		VARCHAR(200)
Product_Category	NOT NULL	VARCHAR(10)
Max_Tenure		Number(2)
Min_Tenure		Number(2)

4. Create the table Loan_Agreement based on the following instance chart.
   Name the table: **LMS_AGREEMENT_DTL**

Name	Null?	Type	Description
Agreement_ID	NOT NULL	VARCHAR(20)	
Lessee_ID	NOT NULL	VARCHAR(100)	
Tenure		NUMBER(2)	
ROI		NUMBER(3,1)	
Loan_Amount		NUMBER(10,2)	
Repayment_Frequency		VARCHAR(2)	M – Monthly Y – Yearly Q – Quarterly HY – Half Yearly
Loan_Disbursal_Date		DATE	
Status		VARCHAR(10)	Pending/Approved/Rejected /Active/Closed
Product_Code	NOT NULL	VARCHAR(20)	

5. Create the table EMI_Schedule.
   Name the table: **LMS_REPAYSCH_DTL**

Name	Null?	Type	Description
Agreement_id	NOT NULL	Varchar(20)	
PropInstID	NOT NULL	Number(8)	Unique Self generated Sequence Number of the each instalment
Installment_Amount	NOT NULL	Number(8,2)	
Installment_Number	NOT NULL	Number(3)	
Principal_Component	NOT NULL	Number(8,2)	
Interest_Component	NOT NULL	Number(8,2)	
Balance_Principal_Amount	NOT NULL	Number(8,2)	
Penalty_Charges		Number(8,2)	
Installment_Due_Date	NOT NULL	Date	

6. Create the table Transaction_Type.
   Name the table: **LMS_TXNTYPE_M**

Name	Null?	Type	Description
Txn_Type	Not Null	VARCHAR(20)	Values – Installment/ LPP
Description		VARCHAR(100)	

7. Create the table Advice.
   Name the table: **LMS_TXN_ADVICE_DTL**

Name	Null?	Type	Description
Txn_Advice_ID	Not Null	Number(8)	unique identification number generated by the system for each advice
Adviceamt	Not Null	Number(8,2)	
AdviceDate	Not Null	Date	
Case_Id	Not Null	Varchar(20)	
TxnID	Not Null	Number(8)	
TxnType	Not Null	Varchar(20)	

8. Create the table Payment.
   Name the table: **LMS_CHEQUE_DTL**

Name	Null?	Type	Description
Cheque_Id	NOT NULL	NUMBER(8)	Unique system identifier
Payment_Mode	NOT NULL	VARCHAR(1)	Mode of payment of instrument. Applicable values are C - Cash Q - Cheque D - Draft T - Fund Transfer
Cheque_Num	NOT NULL	VARCHAR(50)	Instrument number (in case of cheque, it will be cheque number and in case of Fund Transfer it will be Transfer Account Number)
Cheque_Date		DATE	
Cheque_Amount		NUMBER(10,2)	
Drawn_On_Bank		VARCHAR(50)	
Deposit_Date		DATE	
Status		VARCHAR(1)	Status of the cheque C - Realized D - Deposit X - cancelled B - bounced

9. Create the table Receipt_Allocation.
   Name the table: **LMS_PAYMENT_DTL**

Name	Null?	Type	Description
Payment_ID	NOT NULL	Number(8)	Unique system identifier
Cheque_ID	NOT NULL	Number(8)	Unique system identifier
Payment_Date	NOT NULL	Date	
Status	NOT NULL	Varchar(1)	Status of the Payment C – Completed N – Not Completed
Txn_Advice_Id	NOT NULL	Number(8)	

10. Confirm that all the tables are in Data Dictionary

11. Alter the tables to add the primary key as per the below given chart

Primary Key Reference Name	Table Name	Column Name
LMS_CUSTOMER_M_PK	LMS_CUSTOMER_M	Customer_ID
LMS_PRODUCT_CAT_M_PK	LMS_PRODUCT_CATEGORY_M	Category_ID
LMS_PRODUCT_M_PK	LMS_PRODUCT_M	Product_Code
LMS_AGREEMENT_DTL_PK	LMS_AGREEMENT_DTL	Agreement_ID
LMS_REPAYSCH_DTL_PK	LMS_REPAYSCH_DTL	PropInstID
LMS_TXNTYPE_M_PK	LMS_TXNTYPE_M	TxnType
LMS_TXN_ADVICE_DTL_PK	LMS_TXN_ADVICE_DTL	Txn_Advice_ID
LMS_CHEQUE_DTL_PK	LMS_CHEQUE_DTL	Cheque_ID
LMS_PAYMENT_DTL_PK	LMS_PAYMENT_DTL	Payment_ID

12. Alter the tables to add the foreign keys as per the below given chart

Foreign Key Reference Name	Table Name	Column Name	Referenced Table	Referenced Column Name
LMS_Product_Cat_FK	LMS_Product_M	Product_Category	LMS_Product_Category_M	Category_ID
LMS_Agreement_Dtl_FK1	LMS_Agreement_DTL	Lessee_ID	LMS_Customer_M	Customer_ID
LMS_Agreement_DTL_FK2	LMS_Agreement_DTL	Product_code	LMS_Product_M	Product_Code
LMS_Repaysch_Dtl_FK	LMS_Repaysch_Dtl	Agreement_Id	LMS_Agreement_DTL	Agreement_id
LMS_TXN_ADVICE_DTL_FK1	LMS_TXN_ADVICE_DTL	TxnType	LMS_TXNTYPE_M	Txn_Type
LMS_TXN_ADVICE_DTL_FK2	LMS_TXN_ADVICE_DTL	Case_Id	LMS_Agreement_DTL	Agreement_id
LMS_TXN_ADVICE_DTL_FK2	LMS_TXN_ADVICE_DTL	TxnID	LMS_REPAYSCH_DTL	PropInstID
LMS_Payment_Dtl_FK1	LMS_Payment_DTL	Cheque_ID	LMS_CHEQUE_DTL	Cheque_id
LMS_Payment_Dtl_FK2	LMS_Payment_DTL	Txn_Advice_Id	LMS_TXN_ADVICE_DTL	Txn_Advice_id

13. Modify the Customer table to allow for longer Customer Last Names. Confirm your modification

Name	Null?	Type
Customer_ID	NOT NULL	VARCHAR(20)
First_Name	NOT NULL	VARCHAR(20)
Last_Name		**VARCHAR(30)**
Gender	NOT NULL	CHAR(1)
Date_Of_Birth	NOT NULL	DATE
Contact_Number	NOT NULL	VARCHAR(10)
Email_Address	NOT NULL	VARCHAR(40)
Monthly_Income	NOT NULL	NUMBER(8,2)

Profession		VARCHAR(40)
Total_Monthly_Expense	NOT NULL	NUMBER(8,2)
Designation		VARCHAR(40)
Company_Name		VARCHAR(40)

14. Create the table Customer2 based on the structure of Customer table. Include only the Customer_id, first_name, email_address, profession, monthlyIncome. Name the columns in your new table as ID, NAME, EMAIL, PROFESSION, INCOME.

Name	Null?	Type
ID	NOT NULL	VARCHAR(20)
NAME	NOT NULL	VARCHAR(20)
EMAIL	NOT NULL	VARCHAR(40)
PROFESSION		VARCHAR(40)
INCOME		NUMBER(8,2)

15. Drop the table Customer2.

16. Query the recycle bin to see whether the table is present.

17. Undrop the table Customer2.

18. Drop the column FIRST_NAME from the Customer2 table. Confirm your modification by checking the description of the table.

DML (Data Manipulation Language)

You have been given a excel file containing the data. Use the same data to fill in the details as mentioned below:

1. Create an INSERT statement to add the first row of the data in the Customer table being created. Do not list the columns in the INSERT Clause. Do not enter all rows yet.

2. Populate the CUSTOMER table with the second row of sample data from the preceeding list. This time, list the columns explicitly in the INSERT clause.

3. Confirm your addition to the table.

4. Write an insert statement in a dynamic reusable script file named loadcust.sql to load rows into the CUSTOMER table. Save this script to a file named lab_01_04.sql.

5. Populate the table with the next 10 rows of sample data by running the insert statement in the script that you created in Step 4.

6. Populate the table with the remaining rows of sample data using the excel file.

7. Confirm your additions to the table

8. Make the data additions permanent.

9. In the same manner, insert the rows in the rest of the tables.

10. Change the last name of the Customer 3 to ‘Sharma’.

11. Change the Profession, designation and Company Name of Customer 4 to ‘Service’ , ‘Asst Manager’ , ‘Wipro’ respectively.

12. Delete the customer whose monthly income is less than 40000.

13. Delete the Loans from LoanAgreement table whose disbursal date is before 2000.

14. Commit all pending changes

Control Data Transactions

15. In the Customer table, insert two more rows from the sample data listed in Step by using the statements in the script that you created in Step. Run the statement in the script

16. Mark an intermediate point in the processing of the transaction

17. Empty the entire table

18. Confirm that the table is empty

19. Discard the most recent DELETE without discarding the earlier INSERT operations

20. Confirm that the new row is still intact

21. Make the data addition permanent.