Statement of Purpose for United Loans Database Distribution

Badou Kossonou, Hanif Lumsden, Ibrahim Moudio

University of Maryland Global Campus

DBST663

Dr. Gonzalez

03/30/2022

Table of Contents

Introduction

Scenario

The local student loaning service named United Loans to remain competitive with other loaning businesses, it is opening branches in other parts of the region. Management of the other branches need to view reports regarding each branch; employees need to know and share information regarding transactions and customers at all of the branches. This is particularly important when students want to access any United Loans entity they want and there not be an issue when pulling up their accounts. To satisfy customers, the same bank accounts and services are to be at all locations. Stakeholders are concerned that the data cannot be shared and accessed uniformly. As a result, United Loan hires contractors, DBST663 Group 5, to structure data for both branches such that it can be shared by all. DBST663 Group 5 proposes a distributed database solution. The design and implementation of a distributed banking database is valuable for reasons stated.

Limitations & Assumptions

The project is done within the constraints of Oracle SQL Developer on a virtual machine via UMGC virtual lab. A full banking database containing real values will not be installed, only hypothetical ones to fit the scenario. Distributed database evaluation will be done to the extent of testing. There are only two databases to which the data can be distributed: DBST 663a and DBST 663b. Only five views are created for the project. It is assumed that employees at United Loans will be given administrative privileges to utilize the database link.

Design Decision

Design Reference and Layout

The webpage *Construct an ER diagram for the Banking System in DBMS?* (Tutorials Point, 2021) gives a framework to construct an entity relationship diagram (ERD) for a banking system in a relational database and was used as a design reference to construct the project ERD. There are eight entities for the database: the Student entities are the customers who have a loan taken out with United Loan and therefore have an account as shown in the account entity; this account entity provides information on the loan for each respective student, list of transactions, and the branch associated with said account; the bank employee entity belongs to the branch entity since workers for the company belong to particular branches, all of this information has ties to a location entity in zip code which informs the student address entity. The entity relationship diagram can be altered if requirements need changing.

Assumptions and Security Measures

Student's proof of loan will be validated by their account and loan id associated with it. Back-end administrators can access data from both databases at anytime.

Database Management System

Oracle Database 19c 12.2.0.1 64-bit running on Windows 10 with Oracle SQL developer 19.1.0.094 Build 094.2042 to design the database. This database solution is chosen due to familiarity with Oracle SQL language and the features outlined in the Statement of Work section as it will satisfy the needs of the client for a reliable and stable database. The design of the database itself is structured with the goal in mind of being simple for necessary data definition or manipulation languages to be utilized as seen fit to add new fields and update information in the database.

Statement of Work

Overview

The database will be for the loaning service United Loans so that information can be distributed in an organized manner while being able to be accessed quickly. The company identified a need for a distributed database due to their growing regional expansion. United Loans, using Oracle database, will have their data distributed to respective regions. The following is a statement of purpose to showcase the methodology used to distribute the data. Oracle is a reliable and scalable solution that will satisfy both front-end and back-end user satisfaction and productivity in a cost-effective manner.

Objective of Database Project

The goal is to design and implement a data distribution plan and execution to organize information associated with United Loans. Said goal is to be accomplished in three months to complete the research, design, code, and implementation. The design process will initiate following the purpose of work including research, primary and foreign key specification, refining and normalization, and organize the required information in the form of an ERD and logical model.

Project Scope

The scope of the system will be used as a database solution for the establishment of a distributed data plan. After United Loans approves this statement, distribution of their actual data will be initiated using the Oracle tools and functions outlined in this document. Work is done with the purpose of employees being able to retrieve distributed data in an efficient manner. To ensure efficiency, explain and timing commands are ran. In-scope work will entail:

- Statement of Purpose

- Requirement Definition
- Entity-Relationship Diagram and Business Rules
- Sample Data Definition and Manipulation Language
- Data Distribution Plan
- Fragmentation and Views Queries for quick retrieval
- Explain and Timing Queries to expose performance

The out-of-scope work will entail:

- Installation of Oracle database for all branches and locations.
- Training employees on administration tasks.

Database Goals, Expectations, and Deliverables

The goal will be for information to be distributed and accessed quickly. It is expected that the test data will imitate actual data for United Loans. Retrieval time is expected to be a bit slow considering the amount of JOINs used in the views queries. The relations will be mapped out using Oracle Data Modeler and developed using Oracle SQL Developer application. Deliverables will include this project work plan, ERD, distribution plan and strategy, data definition language, and views and explain plan SQL query scripts to prove distribution functionality.

Database Benefits

United Loans will have a functional distributed database. Oracle is a highly reliable server regardless of licensing costs. System downtime and general expenditure is lowered with a high performance and backup/recovery necessary for constant updating (Cherry Road Consultancy Team, 2021). This allows for database link queries to be ran quickly.

Project Hardware / Software Tools

Diagramming Tools Identified

Oracle SQL Developer Data Modeler is utilized. Oracle SQL Developer Data Modeler 19.1.0.081 Build 081.0911 in Windows 10 is used to create the ERD and logical model.

Office Productivity

Google Docs on both Firefox 98.0.2 (64-bit) and Google Chrome 99.0.4844.82 (Official Build) (64-bit) to create the report, running on Windows 10.

Database and Access Method Identified

Oracle Database 19c is used. Oracle SQL Developer version 19.1.0.094 build 094.2042 running on Windows 10 is used as the medium for the database management system. The DDL is created after creating the ERD and logical model in Oracle SQL Developer Data Modeler.

Client Access

Information is accessed by all employees regardless of the branch they belong to by running distributed database queries and the views created.

SQL Usage and Style Guide

The back-end user is granted privileges for usage of the DDL and DML:

- Neatly formatted code written out in a .txt file and transferred to the Oracle SQL developer workspace.
- Usage of comments using "-" for single line comments in the DDL and DML to communicate the reason behind each code.
- Naming conventions to avoid errors.
- Capitalized DDL and DML queries for visual clarity.

Requirements Definition

Relationship, Cardinality, and Business Rules

- Relationship: 1:1: Student and Account
    - Cardinality/Business Rules: A student can be associated to one and only one account. An account can be associated to one and only one student
- Relationship: 1:1: Student and StudentAddress
    - Cardinality/Business Rules: A student can have one and only one address. A student address can have one and only one student.
- Relationship:1:M: Account and BankEmployee
    - Cardinality/Business Rules: An account can be verified by one and only Bank employee. A bank employee verifies zero, one or many accounts.
- Relationship: 1:M: Account and Branch
    - Cardinality/Business Rules: One and only one account belongs to A branch. A branch belongs to one or many accounts.
- Relationship:1:M: Branch and Zipcode
    - Cardinality/Business Rules: A branch can be located at one and only zip code. A zip code can locate zero, one or many branches.
- Relationship: 1:M: Student and Loan
    - Cardinality/Business Rules: A student borrows zero, one or many loans. A loan can be borrowed by one and only one student.
- Relationship: 1:M: Account and Loan
    - Cardinality/Business Rules: An Account may be for zero, one or many loans. A loan is for one and only one account.

- Relationship: 1:M: Branch and BankEmployee
    - Cardinality/Business Rules: A branch can have one or many employees working. An employee can work in one and only one branch.
- Relationship:1:M: BankEmployee and BankEmployee
    - Cardinality/Business Rules: An employee can be managed by one and only one manager. A manager manages zero, one or many employees.
- Relationship: 1:M: Transaction and Account
    - Cardinality/Business Rules: An account receives zero, one or many transactions. A transaction can be received by one and only one account.
- Relationship:1:M: StudentAddress and Zipcode
    - Cardinality/Business Rules: A student address is at one and only one zip code. A given zip code can be for zero, one or many students addresses

Note: Relationship:1:M: BankEmployee and BankEmployee is a recursive relationship.

Entity and Attribute Description

- **Student: Student borrowing loan**
    - *StudentID : Primary Key – assigned unique student Identification*
    - *First_Name: First name of student who borrows loan*
    - *Last_Name: Last Name of Student who borrows loan*
    - *DateofBirdth: Date of birth of student who borrows loan*
    - *PlaceOfBirth: Place of Birth of student who borrows loan*
    - *Social_Security_Number: Social Security Number of Student who borrows loan*
    - *Phone: phone contact of student who borrows the loan*
    - *Email: Email of student who borrows loan*

- *StudentAddress_ID: Foreign key from student address entity establishing their relationship*
- *AccountID: Foreign key from Account entity establishing their relationship*

- **Account: Student Loan Account**
  - *AccountID: Primary Key – assigned unique account Identification*
  - *AcountOpeningDate : Student loan account opening date*
  - *ACountInitialBalance: Student Initial Amount Borrowed*
  - *AccountCurrentBalance: Student current balance*
  - *AccountStatus : Student Account Status*
  - *EmpID: Foreign key from BankEmplyee entity establishing their relationship*
  - *BranchID: Foreign key from Branch entity establishing their relationship*
- **Branch: The bank loan Branch**
  - *BranchID: Primary Key – assigned unique branch Identification*
  - *BranchName: The name of Bank's Branch*
  - *BranchManagerID: The Idenfier of the branch manager*
  - *Zipcode: Foreign key from zipcode entity establishing their relationship*
- **Loan: The loan borrowed by the student**
  - *LoanID: Prymary Key – assigned unique loan Identification*
  - *Name: Loan Name*
  - *PaymentMode: Loan Payment Mode*
  - *Date_Issued: Loan issued date*
  - *Balance: Loan Initial Amount*
  - *Interest: Loan interest based on the amount borrowed*

        - *StudentID: Foreign key from student entity establishing their relationship*
        - *AccountID: Foreign key from Account entity establishing their relationship*
    - **<u>BankEmployee: The employees of the bank</u>**
        - *EmpID: Prymary Key – assigned unique BankEmployee Identification*
        - *First_Name: Employee first name*
        - *Last_Name: Employee last name*
        - *DateOfBirth: Date of Birth*
        - *HireDate: Employee hire date*
        - *Role: employee job title*
        - *Phone: employee phone contact*
        - *Email: employee email*
        - *BranchID: Foreign key from branch entity establishing their relationship*
        - *EmpID: Foreign key from BankEmployee entity establishing Recursive relationship*
    - **<u>Transaction: The transactions of the loan different payments</u>**
        - *TransactionID: Prymary Key – assigned unique Transaction Identification*
        - *Amount: Transaction Amount*
        - *DateOfTransaction: Transaction Date*
        - *TransactionTime: Time of Transaction*
        - *Balance: Transaction Set Balance*
        - *TransactionType: The type of transaction (Online payment, Check, Money Order)*
        - *AccountID: Foreign key from Account entity establishing their relationship*

- **Zipcode: The zipcode representing the branch and students' locations**
    - *Zipcode: Prymary Key – assigned unique zipcode Identification*
    - *City: City of the zipcode*
    - *State: State of the zipcode*
    - *Country: Country of the zipcode*
- **StudentAddress : The student addresses**
    - *StudentAddressID: Prymary Key – assigned unique student address Identification*
    - *StudentAddress: The Address of the student borrowing loan*
    - *Zipcode: Foreign key from zipcode entity establishing their relationship*

Assumptions and Special Considerations

These relationships and entities use *Construct an ER diagram for the Banking System in DBMS?* (Tutorials Point, 2021).

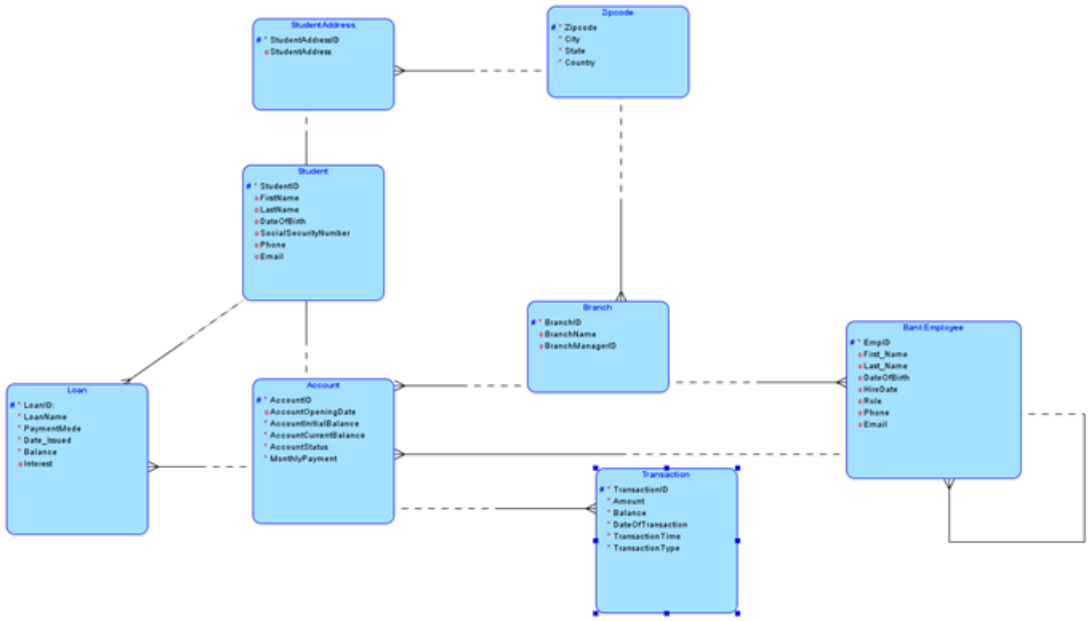
Database Design

Entity-Relationship Diagram and Relational Schema

Using Oracle SQL Developer Data Modeler program the following diagrams are created:

Figure 1

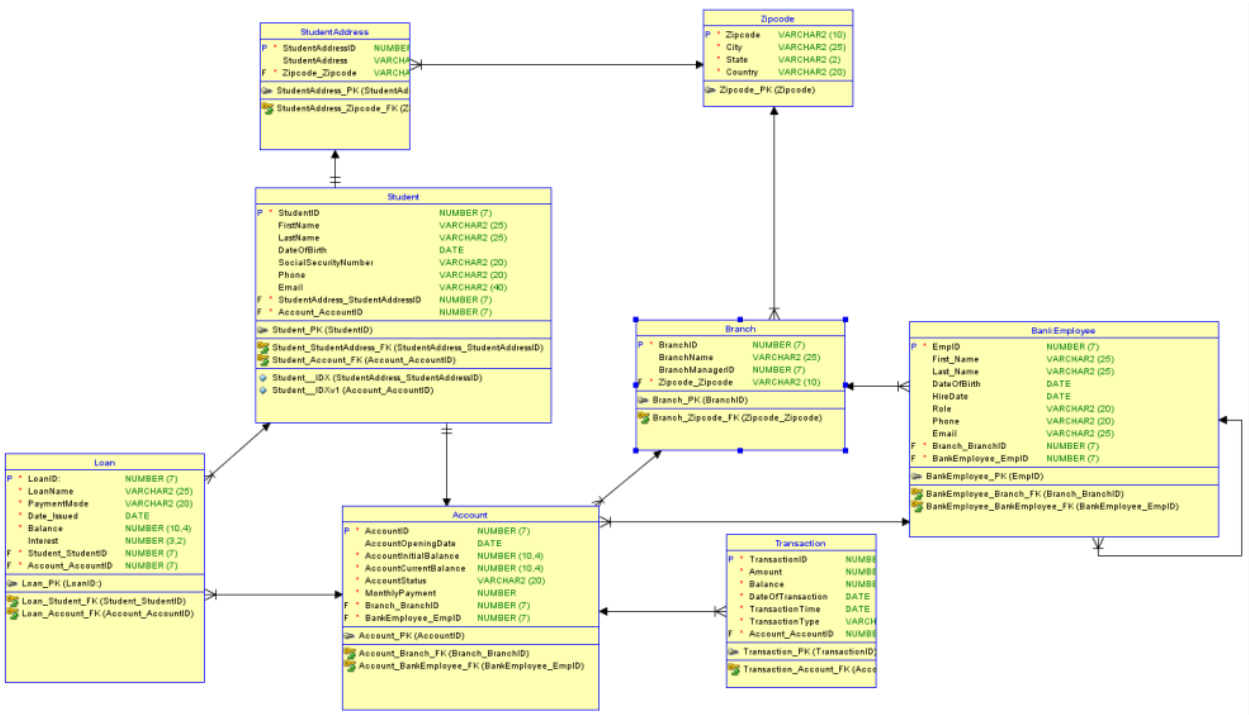
Logical Model



Note. The following entities are present: Student, Account, Branch, Loan, BankEmployee, Transaction, Zipcode, StudentAddress.

Figure 2

Normalized Relations



Note. The following entities are present: Student, Account, Branch, Loan, BankEmployee, Transaction, Zipcode, StudentAddress.

Database Distribution Plan

There will be a data distribution between geographically different bank loan servicing database links. The Entity Relationship Diagram as shown in **figure 2** is used. The Branch, Student Address, and Zipcode tables will be fragmented by the Maryland and North Carolina entites. The reason being is that the branch table contains all branches for the bank company stretching across all databases and partitioning data by location makes sense. Note, this is not representative of the actual data for United Loans. For the two databases: DBST663a and DBST663b, there will be a split in the Branch, Student Address, and Zipcode table two ways as horizontal fragments. The fragmentation is described:

$$TABLE_1 = {}^{S}TABLEID \leq "ID_{M/2}"^{(TABLEID)} \quad \textbf{(1)}$$

$$TABLE_2 =^{S} TABLEID > "ID_{M/2}"^{(TABLEID)} \quad \textbf{(2)}$$

Where M is the total amount of rows in the database. This equation assumes the location-based fragmentation to be split evenly. The fragmentation strategy will be for the database to be split by location and branch for any data populated across stated entities is to be distributed accordingly based on the entities in question:

Figure 3

Example of the split of database data of the BRANCH entity

BRANCHID	BRANCHNAME	BRANCHMANAGERID	ZIPCODE
ID_1	Lorem	ManagerID1	10001
ID_2	Ipsum	ManagerID2	10002
ID_3	Dolor	ManagerID3	10003
ID_4	Sit	ManagerID4	10004
…	…	…	…
ID_{M-3}	Anim	$ManagerID_{N-3}$	AAAAA-3
ID_{M-2}	Id	$ManagerID_{N-2}$	AAAAA-2
ID_{M-1}	Est	$ManagerID_{N-1}$	AAAAA-1
ID_M	Laborum	$ManagerID_N$	AAAAA

Note. AAAAA is the final number of zip codes in the BRANCH entity. The data presented in this figure does not actually represent the data of the BRANCH entity but a hypothetical one.

For this example, there will be only two rows created and split between two databases to test.

Process

Database Link and Distribution Queries

The data will be split using the strategy described in equations **(1)** and **(2)**. The database link syntax is 'bank_a_b'. A query is run:

```
CREATE DATABASE LINK bank_a_b
      CONNECT TO
            DBST_USER
      IDENTIFIED BY
            SecurePassword
      USING
            'dbst663b';

SELECT * FROM DUAL@bank_a_b;
```

Tables will be created to test the database link and distribution in full. The script is written.

```
CREATE TABLE branch (
    branchid        NUMBER(7) NOT NULL,
    branchname      VARCHAR2(25),
    branchmanagerid NUMBER(7),
    zipcode_zipcode VARCHAR2(10) NOT NULL
);
```

To test distribution and its efficacy a few test DML queries are run to be deleted:

```
INSERT INTO zipcode (
    zipcode, city, state, country
    ) VALUES (
        '30302', 'Atlanta', 'GA', 'United States'
        );

INSERT INTO zipcode (
```

```
    zipcode, city, state, country
    ) VALUES (
        '21201', 'Baltimore', 'MD', 'United States'
        );

INSERT INTO branch (
    branchid, branchname, branchmanagerid, zipcode_zipcode
) VALUES (
        1000001, 'United Capital Atlanta', 2000024, '30302'
        );

INSERT INTO branch@bank_a_b (
    branchid, branchname, branchmanagerid, zipcode_zipcode
) VALUES (
        1000006, 'United Capital Baltimore ', 2000045, '21201',
);
```

A view is created to see both database and to validate the connection:

```
CREATE OR REPLACE VIEW all_branches AS
    SELECT * FROM branch
    UNION
    SELECT * FROM branch@bank_a_b;

SELECT * FROM all_branches;
```

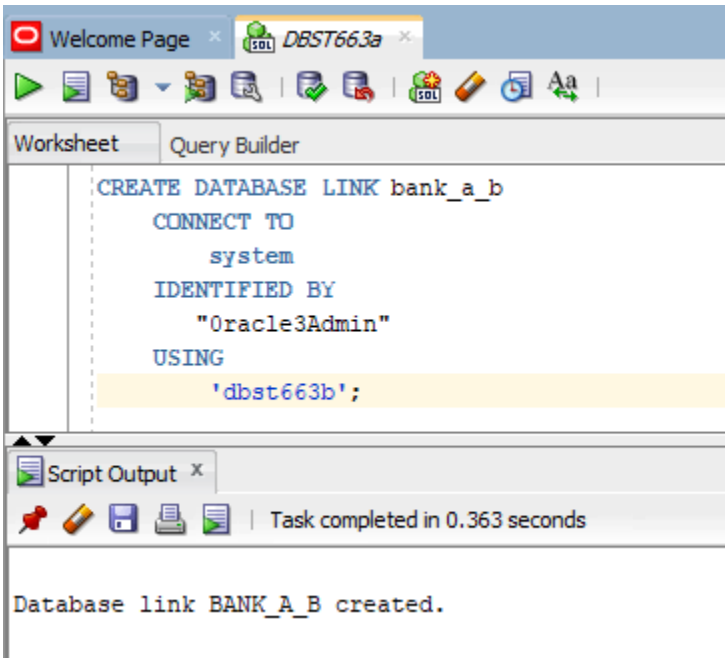
Since this is a test, the sample row data created in this are deleted from the database to be populated in the next DML step.

```
DELETE FROM branch;
```

The following is a screenshot of the actual fragmentations:

Figure 4

Database Link Creation



Data Definition and Manipulation Language Query Source Code

```
--------------------------------------------------------
--  File created - Saturday-March-26-2022
--------------------------------------------------------
--------------------------------------------------------
--  DDL for DB Link BANK_A_B
--------------------------------------------------------

  CREATE DATABASE LINK "BANK_A_B"
   CONNECT TO "DBST_USER" IDENTIFIED BY VALUES ':1'
   USING 'dbst663b';
--------------------------------------------------------
--  DDL for DB Link FR_A_TO_B
--------------------------------------------------------

  CREATE DATABASE LINK "FR_A_TO_B"
   CONNECT TO "DBST_USER" IDENTIFIED BY VALUES ':1'
   USING 'dbst663b';
--------------------------------------------------------
--  DDL for Sequence ACCOUNT_SEQ
--------------------------------------------------------

   CREATE SEQUENCE  "DBST_USER"."ACCOUNT_SEQ"  MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 41 CACHE 20 NOORDER  NOCYCLE
```

```
NOKEEP  NOSCALE  GLOBAL ;
--------------------------------------------------------
--  DDL for Sequence BANKEMPLOYEE_SEQ
--------------------------------------------------------

   CREATE SEQUENCE  "DBST_USER"."BANKEMPLOYEE_SEQ"  MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 61 CACHE 20 NOORDER  NOCYCLE
NOKEEP  NOSCALE  GLOBAL ;
--------------------------------------------------------
--  DDL for Sequence BRANCH_SEQ
--------------------------------------------------------

   CREATE SEQUENCE  "DBST_USER"."BRANCH_SEQ"  MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 41 CACHE 20 NOORDER  NOCYCLE
NOKEEP  NOSCALE  GLOBAL ;
--------------------------------------------------------
--  DDL for Sequence LOAN_SEQ
--------------------------------------------------------

   CREATE SEQUENCE  "DBST_USER"."LOAN_SEQ"  MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 141 CACHE 20 NOORDER
NOCYCLE  NOKEEP  NOSCALE  GLOBAL ;
--------------------------------------------------------
--  DDL for Sequence STUDENTADDRESS_SEQ
--------------------------------------------------------

   CREATE SEQUENCE  "DBST_USER"."STUDENTADDRESS_SEQ"  MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 41 CACHE 20 NOORDER  NOCYCLE
NOKEEP  NOSCALE  GLOBAL ;
--------------------------------------------------------
--  DDL for Sequence STUDENT_SEQ
--------------------------------------------------------

   CREATE SEQUENCE  "DBST_USER"."STUDENT_SEQ"  MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 281 CACHE 20 NOORDER
NOCYCLE  NOKEEP  NOSCALE  GLOBAL ;
--------------------------------------------------------
--  DDL for Sequence TRANSACTION_SEQ
--------------------------------------------------------

   CREATE SEQUENCE  "DBST_USER"."TRANSACTION_SEQ"  MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT BY 1 START WITH 41 CACHE 20 NOORDER  NOCYCLE
NOKEEP  NOSCALE  GLOBAL ;
--------------------------------------------------------
--  DDL for Table ZIPCODE
--------------------------------------------------------
```

```
  CREATE TABLE "DBST_USER"."ZIPCODE"
   (   "ZIPCODE" VARCHAR2(10 BYTE),
       "CITY" VARCHAR2(25 BYTE),
       "STATE" VARCHAR2(2 BYTE),
       "COUNTRY" VARCHAR2(20 BYTE)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;

   COMMENT ON COLUMN "DBST_USER"."ZIPCODE"."ZIPCODE" IS ' Zipcode of the location
';
   COMMENT ON COLUMN "DBST_USER"."ZIPCODE"."CITY" IS ' City of the zipcode';
   COMMENT ON COLUMN "DBST_USER"."ZIPCODE"."STATE" IS 'State of the zipcode';
   COMMENT ON COLUMN "DBST_USER"."ZIPCODE"."COUNTRY" IS ' Country of the zipcode ';
--------------------------------------------------------
--  DDL for Table STUDENTADDRESS
--------------------------------------------------------

  CREATE TABLE "DBST_USER"."STUDENTADDRESS"
   (   "STUDENTADDRESSID" NUMBER(7,0),
       "STUDENTADDRESS" VARCHAR2(40 BYTE),
       "ZIPCODE_ZIPCODE" VARCHAR2(10 BYTE)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;

   COMMENT ON COLUMN "DBST_USER"."STUDENTADDRESS"."STUDENTADDRESSID" IS ' Student
address Identifier ';
   COMMENT ON COLUMN "DBST_USER"."STUDENTADDRESS"."STUDENTADDRESS" IS 'The Address
of the student borrowing loan ';
--------------------------------------------------------
--  DDL for Table BRANCH
--------------------------------------------------------

  CREATE TABLE "DBST_USER"."BRANCH"
   (   "BRANCHID" NUMBER(7,0),
       "BRANCHNAME" VARCHAR2(25 BYTE),
       "ZIPCODE_ZIPCODE" VARCHAR2(10 BYTE)
   ) SEGMENT CREATION IMMEDIATE
```

```
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;

   COMMENT ON COLUMN "DBST_USER"."BRANCH"."BRANCHID" IS ' Bank Identifier branch ';
   COMMENT ON COLUMN "DBST_USER"."BRANCH"."BRANCHNAME" IS 'The name of Bank's
Branch';
--------------------------------------------------------
--  DDL for Table BANKEMPLOYEE
--------------------------------------------------------

  CREATE TABLE "DBST_USER"."BANKEMPLOYEE"
   (   "EMPID" NUMBER(7,0),
       "FIRST_NAME" VARCHAR2(25 BYTE),
       "LAST_NAME" VARCHAR2(25 BYTE),
       "HIREDATE" DATE,
       "ROLE" VARCHAR2(20 BYTE),
       "PHONE" VARCHAR2(20 BYTE),
       "EMAIL" VARCHAR2(25 BYTE),
       "BRANCH_BRANCHID" NUMBER(7,0)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;

   COMMENT ON COLUMN "DBST_USER"."BANKEMPLOYEE"."EMPID" IS ' Employee Identifier ';
   COMMENT ON COLUMN "DBST_USER"."BANKEMPLOYEE"."FIRST_NAME" IS ' Employee first
name ';
   COMMENT ON COLUMN "DBST_USER"."BANKEMPLOYEE"."LAST_NAME" IS 'Employee last name
';
   COMMENT ON COLUMN "DBST_USER"."BANKEMPLOYEE"."HIREDATE" IS 'Employee hire date
';
   COMMENT ON COLUMN "DBST_USER"."BANKEMPLOYEE"."ROLE" IS ' employee job title ';
   COMMENT ON COLUMN "DBST_USER"."BANKEMPLOYEE"."PHONE" IS 'employee phone
contact';
   COMMENT ON COLUMN "DBST_USER"."BANKEMPLOYEE"."EMAIL" IS 'employee email ';
--------------------------------------------------------
--  DDL for Table ACCOUNT
--------------------------------------------------------

  CREATE TABLE "DBST_USER"."ACCOUNT"
```

```
   (   "ACCOUNTID" NUMBER(7,0),
       "ACCOUNTOPENINGDATE" DATE,
       "ACCOUNTINITIALBALANCE" NUMBER(10,4),
       "ACCOUNTCURRENTBALANCE" NUMBER(10,4),
       "ACCOUNTSTATUS" VARCHAR2(20 BYTE),
       "MONTHLYPAYMENT" NUMBER,
       "BRANCH_BRANCHID" NUMBER(7,0),
       "BANKEMPLOYEE_EMPID" NUMBER(7,0)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;

   COMMENT ON COLUMN "DBST_USER"."ACCOUNT"."ACCOUNTID" IS 'Account Identifier';
   COMMENT ON COLUMN "DBST_USER"."ACCOUNT"."ACCOUNTOPENINGDATE" IS 'Student loan
account opening date ';
   COMMENT ON COLUMN "DBST_USER"."ACCOUNT"."ACCOUNTCURRENTBALANCE" IS 'Student
current balance ';
   COMMENT ON COLUMN "DBST_USER"."ACCOUNT"."ACCOUNTSTATUS" IS 'Student Account
Status ';
   COMMENT ON COLUMN "DBST_USER"."ACCOUNT"."MONTHLYPAYMENT" IS 'The monthly payment
';
--------------------------------------------------------
--  DDL for Table TRANSACTION
--------------------------------------------------------

  CREATE TABLE "DBST_USER"."TRANSACTION"
   (   "TRANSACTIONID" NUMBER(7,0),
       "AMOUNT" NUMBER(10,4),
       "DATEOFTRANSACTION" DATE,
       "TRANSACTIONTYPE" VARCHAR2(20 BYTE),
       "ACCOUNT_ACCOUNTID" NUMBER(7,0)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;

   COMMENT ON COLUMN "DBST_USER"."TRANSACTION"."TRANSACTIONID" IS 'Transaction
Identifier ';
   COMMENT ON COLUMN "DBST_USER"."TRANSACTION"."AMOUNT" IS ' Transaction Amount';
   COMMENT ON COLUMN "DBST_USER"."TRANSACTION"."DATEOFTRANSACTION" IS 'Transaction
```

```
Date and Time';
   COMMENT ON COLUMN "DBST_USER"."TRANSACTION"."TRANSACTIONTYPE" IS 'The Type of
the transaction ';
--------------------------------------------------------
--  DDL for Table STUDENT
--------------------------------------------------------

  CREATE TABLE "DBST_USER"."STUDENT"
   (   "STUDENTID" NUMBER(7,0),
       "FIRSTNAME" VARCHAR2(25 BYTE),
       "LASTNAME" VARCHAR2(25 BYTE),
       "DATEOFBIRTH" DATE,
       "SOCIALSECURITYNUMBER" VARCHAR2(20 BYTE),
       "PHONE" VARCHAR2(20 BYTE),
       "EMAIL" VARCHAR2(40 BYTE),
       "STUDENTADDRESS_STUDENTADDRESSID" NUMBER(7,0),
       "ACCOUNT_ACCOUNTID" NUMBER(7,0)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;

   COMMENT ON COLUMN "DBST_USER"."STUDENT"."STUDENTID" IS 'student Identifier';
--------------------------------------------------------
--  DDL for Table LOAN
--------------------------------------------------------

  CREATE TABLE "DBST_USER"."LOAN"
   (   "LOANID" NUMBER(7,0),
       "LOANNAME" VARCHAR2(25 BYTE),
       "DATE_ISSUED" DATE,
       "BALANCE" NUMBER(10,4),
       "INTEREST" NUMBER(7,4),
       "STUDENT_STUDENTID" NUMBER(7,0),
       "ACCOUNT_ACCOUNTID" NUMBER(7,0)
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;

   COMMENT ON COLUMN "DBST_USER"."LOAN"."LOANID" IS ' The Loan Identifier ';
```

```
   COMMENT ON COLUMN "DBST_USER"."LOAN"."LOANNAME" IS ' Loan Name ';
   COMMENT ON COLUMN "DBST_USER"."LOAN"."DATE_ISSUED" IS ' Loan issued date ';
   COMMENT ON COLUMN "DBST_USER"."LOAN"."BALANCE" IS ' Loan Initial Amount ';
   COMMENT ON COLUMN "DBST_USER"."LOAN"."INTEREST" IS 'the interest of the loan ';
--------------------------------------------------------
--  DDL for View ALL_EMP_VIEW
--------------------------------------------------------

  CREATE OR REPLACE FORCE EDITIONABLE VIEW "DBST_USER"."ALL_EMP_VIEW" ("ENO",
"ENAME", "TITLE", "DATABASE") AS
  SELECT "ENO","ENAME","TITLE","DATABASE"
FROM emp
UNION
SELECT "ENO","ENAME","TITLE","DATABASE"
FROM emp@fr_a_to_b
;
REM INSERTING into DBST_USER.ZIPCODE
SET DEFINE OFF;
Insert into DBST_USER.ZIPCODE (ZIPCODE,CITY,STATE,COUNTRY) values
('28262','charlotte','NC','United States');
Insert into DBST_USER.ZIPCODE (ZIPCODE,CITY,STATE,COUNTRY) values
('28213','charlotte','NC','United States');
Insert into DBST_USER.ZIPCODE (ZIPCODE,CITY,STATE,COUNTRY) values
('36104','Montgomery','AL','United States');
Insert into DBST_USER.ZIPCODE (ZIPCODE,CITY,STATE,COUNTRY) values
('46225','Indianapolis','IN','United States');
Insert into DBST_USER.ZIPCODE (ZIPCODE,CITY,STATE,COUNTRY) values
('27601','Raleigh','NC','United States');
Insert into DBST_USER.ZIPCODE (ZIPCODE,CITY,STATE,COUNTRY) values
('85001','Phoenix','AZ','United States');
Insert into DBST_USER.ZIPCODE (ZIPCODE,CITY,STATE,COUNTRY) values
('20588','Waldorf','MD','United States');
Insert into DBST_USER.ZIPCODE (ZIPCODE,CITY,STATE,COUNTRY) values
('20874','Germantown','MD','United States');
REM INSERTING into DBST_USER.STUDENTADDRESS
SET DEFINE OFF;
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (16,'12006 Diploma
Drive','28262');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (17,'9511 Shannon Green
Drive','28213');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (18,'10006 Graduate
Line','28262');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (19,'2321 Mexico Street
```

```
Road','36104');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (20,'1232 School Street
Road','46225');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (21,'1032 Saddle
Center','27601');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (22,'98012 Banks
Street','20588');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (23,'1106 Featherbrook
Road','20874');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (24,'8827 Auburn whisper
Ln','28262');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (25,'3557 Sharon Amity
Rd','20874');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (26,'9711 David Taylor Dr
','85001');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (27,'3805 Tiffany Rose
PI','85001');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (28,'5300 N Tryon
Street','46225');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (29,'Daisy Moore
Ln','36104');
Insert into DBST_USER.STUDENTADDRESS
(STUDENTADDRESSID,STUDENTADDRESS,ZIPCODE_ZIPCODE) values (30,'9331 JW Clay
Blvd','27601');
REM INSERTING into DBST_USER.BRANCH
SET DEFINE OFF;
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values
(16,'Climb Bank Charlotte','28262');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values
(17,'Climb Bank Alabama','36104');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values (18,'Well
Fargo Maryland','20588');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values
(19,'Climb Bank Arizona','85001');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values (20,'Bank
of America NC-C ','28262');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values
```

```
(21,'Climb Bank Waldorf','20588');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values (22,'Fith
Third Bank NC-C ','28213');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values
(23,'Climb Bank Waldorf ','20588');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values
(24,'Climb Bank Maryland','20874');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values
(25,'Climb Bank Waldorf North','20588');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values
(26,'Climb Bank Waldorf South ','20588');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values (27,'Well
fargo Raleigh North','27601');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values (28,'Bank
of America Indiana','46225');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values (29,'Well
Fargo Indiana ','46225');
Insert into DBST_USER.BRANCH (BRANCHID,BRANCHNAME,ZIPCODE_ZIPCODE) values (30,'Well
Fargo Charlotte ','28262');
REM INSERTING into DBST_USER.BANKEMPLOYEE
SET DEFINE OFF;
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(31,'Arthur
','Badou',to_date('22-JAN-09','DD-MON-RR'),'President','704-968-8982','koffiarthur1
1@gmail.com',20);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(32,'Micheal ','Johnson',to_date('13-JAN-10','DD-MON-RR'),'Teller','202-283-8482','
      michaelJ01@yahoo.com',21);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(33,'Pelagie
','Badou',to_date('23-MAY-09','DD-MON-RR'),'Analyst','800-123-8802','badou@hotmail.
com',22);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(34,'Steve
','Mark',to_date('23-MAY-10','DD-MON-RR'),'Manager','919-093-3252','markS@gmail.com
',24);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(35,'John  ','Cury',to_date('04-NOV-11','DD-MON-RR'),'Financial
Analyst','301-342-2352','CuryJohn@yahoo.fr',28);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(36,'Simon
```

```
','roller',to_date('23-MAR-12','DD-MON-RR'),'Manager','704-657-8332','SimonR12@gmai
l.com',21);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(37,'Stephen
','Kumaga',to_date('24-JAN-09','DD-MON-RR'),'Teller','704-837-0923','sefakumaga@gma
il.com',21);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(38,'John
','Miller',to_date('14-APR-13','DD-MON-RR'),'Commercial','980-323-8932','millerJohn
@gmail.com',22);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(39,'Carlos
','Dieggo',to_date('12-AUG-11','DD-MON-RR'),'IT-Manager','401-322-3232','carlosd@gm
ail.com',23);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(40,'Rome ','Atsin',to_date('10-JUL-10','DD-MON-RR'),'developer
','704-319-8302','romeatsin@gmail.com',17);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(41,'Garcia
','Torres',to_date('06-DEC-09','DD-MON-RR'),'Manager','980-335-8382','garcia@gmail.
com',18);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values (42,'
Ibrahim
','Lamine',to_date('03-JUN-09','DD-MON-RR'),'President','980-933-8182','ibrahim@gma
il.com',19);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values (43,'
Ibrahim','Hanif',to_date('10-JAN-20','DD-MON-RR'),'President','301-932-8098','hanif
132@gmail.com',29);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(44,'Roger','Kobena',to_date('06-OCT-21','DD-MON-RR'),'account
specialist','704-476-3232','kobena@gmail.com',23);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(45,'kOUASSI','Ebanian',to_date('28-AUG-21','DD-MON-RR'),'Teller','202-354-0933','k
ouassi@gmail.com',25);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(46,'John','Mickeal',to_date('22-DEC-14','DD-MON-RR'),'Analyst','704-092-3823','mic
kael@gmail.com',22);
```

```
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(47,'Dagnago
','Kiko',to_date('21-JUN-09','DD-MON-RR'),'Analyst','980-322-0923','dagnogokiko@gma
il.com',16);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(48,'Sonia','Morgan',to_date('11-NOV-19','DD-MON-RR'),'Manager','980-329-0932','son
iamorgan@gmail.com',16);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(49,'John','Claude',to_date('15-DEC-09','DD-MON-RR'),'Teller','980-392-2232','johnc
laude@gmail.com',18);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(50,'Toress','Yolonda',to_date('25-JAN-11','DD-MON-RR'),'Commercial','202-329-3862'
,'toressyolanda@gmail.com',27);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values (51,'
Leandre ','Bia',to_date('24-NOV-08','DD-MON-RR'),'Loan
Analyst','980-922-9282','tozanbia@gmail.com',29);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(52,'Sefakor
','Kumaga',to_date('20-JUL-12','DD-MON-RR'),'Teller','704-291-2982','sefakorkumaga@
gmail.com',25);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(53,'Crescentia  ','Akoukum',to_date('10-JAN-16','DD-MON-RR'),'Loan
Analyst','980-323-9920','akoumcres@gmail.com',25);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(54,'Roger ','Blinde',to_date('21-OCT-13','DD-MON-RR'),'IT
Analyst','202-398-8392','blinde@gmail.com',29);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(55,'Mireille
','Badou',to_date('16-DEC-14','DD-MON-RR'),'Analyst','704-827-2628','badoum@gmail.c
om',30);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(56,'Esther
','Smith',to_date('06-JUL-07','DD-MON-RR'),'Developer','980-238-2928','smithesther@
gmail.com',30);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values (57,'
lina ','kungo',to_date('22-JUN-09','DD-MON-RR'),'Loan
```

```
Analyst','704-302-2202','kungo@gmail.com',19);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(58,'Eugene','Atta',to_date('21-JAN-09','DD-MON-RR'),'Loan
Analyst','980-930-1221','attaeugene@gmail.com',21);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(59,'Kinde ','Badou',to_date('22-JAN-11','DD-MON-RR'),'Financal
Analyst','401-329-2022','kinde@gmail.com',21);
Insert into DBST_USER.BANKEMPLOYEE
(EMPID,FIRST_NAME,LAST_NAME,HIREDATE,ROLE,PHONE,EMAIL,BRANCH_BRANCHID) values
(60,'Boni
','jessica',to_date('29-JAN-06','DD-MON-RR'),'Teller','202-918-8352','bonijess@gmai
l.com',21);
REM INSERTING into DBST_USER.ACCOUNT
SET DEFINE OFF;
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(16,to_date('22-JAN-21','DD-MON-RR'),20000,8000,'up to date',300,21,32);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(17,to_date('23-DEC-20','DD-MON-RR'),30000,9000,'up to date',304,21,60);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(18,to_date('10-JAN-18','DD-MON-RR'),26000,26000,'Paid',340,21,37);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(19,to_date('23-JAN-18','DD-MON-RR'),40000,20000,'up to date',400,25,45);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(20,to_date('12-DEC-19','DD-MON-RR'),25000,3000,'late',450,18,49);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(21,to_date('16-DEC-14','DD-MON-RR'),50000,10000,'up to date',330,21,32);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(22,to_date('04-JUN-19','DD-MON-RR'),20000,20000,'Paid',230,25,45);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
```

```
(23,to_date('04-JUL-21','DD-MON-RR'),20000,15000,'up to date',250,21,32);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(24,to_date('04-OCT-19','DD-MON-RR'),26000,7000,'up to date',280,18,49);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(25,to_date('30-JUL-19','DD-MON-RR'),20000,14000,'up to date',130,21,32);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(26,to_date('13-JUN-13','DD-MON-RR'),80000,50000,'up to date',530,21,32);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(27,to_date('04-DEC-10','DD-MON-RR'),200000,80000,'up to date',430,25,45);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(28,to_date('12-JUN-21','DD-MON-RR'),30000,8000,'up to date',400,21,60);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(29,to_date('13-DEC-17','DD-MON-RR'),40000,40000,'Paid',480,21,60);
Insert into DBST_USER.ACCOUNT
(ACCOUNTID,ACCOUNTOPENINGDATE,ACCOUNTINITIALBALANCE,ACCOUNTCURRENTBALANCE,ACCOUNTST
ATUS,MONTHLYPAYMENT,BRANCH_BRANCHID,BANKEMPLOYEE_EMPID) values
(30,to_date('03-NOV-08','DD-MON-RR'),23000,10000,'late',270,21,32);
REM INSERTING into DBST_USER.TRANSACTION
SET DEFINE OFF;
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(11,300,to_date('01-MAR-22','DD-MON-RR'),'Credit Card',16);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(12,304,to_date('04-MAR-22','DD-MON-RR'),'Credit Card',18);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(13,340,to_date('02-MAR-22','DD-MON-RR'),'Debit Card',26);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(14,400,to_date('05-MAR-22','DD-MON-RR'),'Bank',30);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(15,450,to_date('02-MAR-22','DD-MON-RR'),'Debit Card',16);
Insert into DBST_USER.TRANSACTION
```

```
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(16,330,to_date('03-MAR-22','DD-MON-RR'),'Debit Card',18);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(17,230,to_date('10-MAR-22','DD-MON-RR'),'Debit Card',25);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(18,250,to_date('12-MAR-22','DD-MON-RR'),'Debit Card',23);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(19,280,to_date('13-MAR-22','DD-MON-RR'),'Credit Card',16);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(20,130,to_date('11-MAR-22','DD-MON-RR'),'Bank',28);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(21,530,to_date('22-MAR-22','DD-MON-RR'),'Bank',27);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(22,430,to_date('12-MAR-22','DD-MON-RR'),'Credit Card ',28);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(23,400,to_date('22-MAR-22','DD-MON-RR'),'Debit Card',25);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(24,480,to_date('12-MAR-22','DD-MON-RR'),'Bank',28);
Insert into DBST_USER.TRANSACTION
(TRANSACTIONID,AMOUNT,DATEOFTRANSACTION,TRANSACTIONTYPE,ACCOUNT_ACCOUNTID) values
(25,270,to_date('23-MAR-22','DD-MON-RR'),'Bank',26);
REM INSERTING into DBST_USER.STUDENT
SET DEFINE OFF;
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values
(211,'Linda','Keniro',to_date('22-OCT-09','DD-MON-RR'),'564-78-4323','704-413-7310'
,'Linda@gmail.com',16,23);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values
(212,'Arthur','Badou',to_date('18-AUG-11','DD-MON-RR'),'346-48-4873','704-453-7653'
,'Arthur11@gmail.com',23,17);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values (213,'Boris
','Apoh',to_date('22-JAN-09','DD-MON-RR'),'648-47-0932','704-413-7310','Boris@gmail
.com',30,18);
Insert into DBST_USER.STUDENT
```

```
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values
(214,'Margeless','Apoh',to_date('23-MAY-10','DD-MON-RR'),'463-78-9383','704-003-780
9','Margeless@gmail.com',22,27);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values
(215,'Sefakor','Kumaga',to_date('04-NOV-11','DD-MON-RR'),'463-47-9303','704-687-901
2','Sefakor@gmail.com',18,28);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values
(216,'Kouakou','Romaric',to_date('23-MAR-12','DD-MON-RR'),'363-38-3823','704-193-70
07','AtsinK@gmail.com',25,22);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values
(217,'Leandre','Tozan',to_date('10-APR-13','DD-MON-RR'),'364-38-3933','980-786-0219
','LeandreT@yahoo.com',26,21);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values (218,'Mary
','Smith',to_date('12-FEB-10','DD-MON-RR'),'462-32-4323','704-212-0849','Mary@gmail
.com',19,19);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values (219,'John
','Smith',to_date('21-JUN-09','DD-MON-RR'),'274-34-4334','704-212-0849','John
@gmail.com',17,20);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values (220,'David
','Sally',to_date('10-JUL-10','DD-MON-RR'),'342-44-4321','704-968-8921','David@hotm
ail.com',29,29);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values
(221,'Thomas','Sally',to_date('12-JUN-09','DD-MON-RR'),'327-83-9483','704-968-8921'
,'Thomas@yahoo.fr',27,16);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values (222,'
Jones','Sally',to_date('03-JUN-09','DD-MON-RR'),'384-48-3293','704-968-8921','Jones
@gmail.com',21,24);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values
```

```
(223,'Yolanda','Torres',to_date('25-JUL-09','DD-MON-RR'),'844-43-9483','980-333-039
1','Yolanda@gmail.com',20,26);
Insert into DBST_USER.STUDENT
(STUDENTID,FIRSTNAME,LASTNAME,DATEOFBIRTH,SOCIALSECURITYNUMBER,PHONE,EMAIL,STUDENTA
DDRESS_STUDENTADDRESSID,ACCOUNT_ACCOUNTID) values
(224,'Kouassi','John',to_date('12-MAR-07','DD-MON-RR'),'474-43-4830','980-863-0201'
,'Kouassi@gmail.com',24,25);
REM INSERTING into DBST_USER.LOAN
SET DEFINE OFF;
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (93,'student loan',to_date('22-JAN-21','DD-MON-RR'),20000,0.05,211,16);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (94,'student loan',to_date('23-DEC-20','DD-MON-RR'),30000,0.05,212,19);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (95,'student loan',to_date('10-JAN-18','DD-MON-RR'),26000,0.05,213,23);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (96,'student loan',to_date('23-JAN-18','DD-MON-RR'),40000,0.4,214,30);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (97,'student loan',to_date('12-DEC-19','DD-MON-RR'),25000,0.05,215,23);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (98,'student loan',to_date('16-DEC-14','DD-MON-RR'),50000,0.4,216,28);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (99,'student loan',to_date('04-JUN-19','DD-MON-RR'),20000,0.05,217,23);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (100,'student loan',to_date('04-JUL-21','DD-MON-RR'),20000,0.05,218,18);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (101,'student loan',to_date('04-OCT-19','DD-MON-RR'),26000,0.05,219,17);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (102,'student loan',to_date('30-JUL-19','DD-MON-RR'),20000,0.05,220,24);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (103,'student loan',to_date('13-JUN-13','DD-MON-RR'),80000,0.03,221,26);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (104,'student loan',to_date('04-DEC-10','DD-MON-RR'),200000,0.03,222,27);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
```

```
values (105,'student loan',to_date('12-JUN-21','DD-MON-RR'),30000,0.05,223,30);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (106,'student loan',to_date('13-DEC-17','DD-MON-RR'),24000,0.05,224,17);
Insert into DBST_USER.LOAN
(LOANID,LOANNAME,DATE_ISSUED,BALANCE,INTEREST,STUDENT_STUDENTID,ACCOUNT_ACCOUNTID)
values (107,'student loan',to_date('03-NOV-08','DD-MON-RR'),23000,0.05,224,20);
--------------------------------------------------------
--  DDL for Index ACCOUNT_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DBST_USER"."ACCOUNT_PK" ON "DBST_USER"."ACCOUNT"
("ACCOUNTID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index BANKEMPLOYEE_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DBST_USER"."BANKEMPLOYEE_PK" ON "DBST_USER"."BANKEMPLOYEE"
("EMPID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index BRANCH_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DBST_USER"."BRANCH_PK" ON "DBST_USER"."BRANCH" ("BRANCHID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index STUDENTADDRESS_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DBST_USER"."STUDENTADDRESS_PK" ON
"DBST_USER"."STUDENTADDRESS" ("STUDENTADDRESSID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
```

```
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index LOAN_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DBST_USER"."LOAN_PK" ON "DBST_USER"."LOAN" ("LOANID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index STUDENT_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DBST_USER"."STUDENT_PK" ON "DBST_USER"."STUDENT"
("STUDENTID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index STUDENT__IDX
--------------------------------------------------------

  CREATE UNIQUE INDEX "DBST_USER"."STUDENT__IDX" ON "DBST_USER"."STUDENT"
("STUDENTADDRESS_STUDENTADDRESSID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index STUDENT__IDXV1
--------------------------------------------------------

  CREATE UNIQUE INDEX "DBST_USER"."STUDENT__IDXV1"g ON "DBST_USER"."STUDENT"
("ACCOUNT_ACCOUNTID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
```

```
--  DDL for Index TRANSACTION_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DBST_USER"."TRANSACTION_PK" ON "DBST_USER"."TRANSACTION"
("TRANSACTIONID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index ZIPCODE_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "DBST_USER"."ZIPCODE_PK" ON "DBST_USER"."ZIPCODE" ("ZIPCODE")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Procedure HIRE_EMP
--------------------------------------------------------
set define off;

  CREATE OR REPLACE EDITIONABLE PROCEDURE "DBST_USER"."HIRE_EMP" (enum NUMBER) AS
BEGIN
INSERT INTO emp@fr_a_to_b (eno, ename, title, database)
VALUES ('ZI', 'Thang Ngo', 'TA', 'DBST663B');
END;

/
--------------------------------------------------------
--  Constraints for Table ZIPCODE
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."ZIPCODE" MODIFY ("ZIPCODE" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."ZIPCODE" MODIFY ("CITY" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."ZIPCODE" MODIFY ("STATE" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."ZIPCODE" MODIFY ("COUNTRY" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."ZIPCODE" ADD CONSTRAINT "ZIPCODE_PK" PRIMARY KEY
("ZIPCODE")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
```

```
--------------------------------------------------------
--  Constraints for Table STUDENTADDRESS
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."STUDENTADDRESS" MODIFY ("STUDENTADDRESSID" NOT NULL
ENABLE);
  ALTER TABLE "DBST_USER"."STUDENTADDRESS" MODIFY ("ZIPCODE_ZIPCODE" NOT NULL
ENABLE);
  ALTER TABLE "DBST_USER"."STUDENTADDRESS" ADD CONSTRAINT "STUDENTADDRESS_PK"
PRIMARY KEY ("STUDENTADDRESSID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
--------------------------------------------------------
--  Constraints for Table BRANCH
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."BRANCH" MODIFY ("BRANCHID" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."BRANCH" MODIFY ("ZIPCODE_ZIPCODE" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."BRANCH" ADD CONSTRAINT "BRANCH_PK" PRIMARY KEY
("BRANCHID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
--------------------------------------------------------
--  Constraints for Table BANKEMPLOYEE
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."BANKEMPLOYEE" MODIFY ("EMPID" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."BANKEMPLOYEE" MODIFY ("BRANCH_BRANCHID" NOT NULL
ENABLE);
  ALTER TABLE "DBST_USER"."BANKEMPLOYEE" ADD CONSTRAINT "BANKEMPLOYEE_PK" PRIMARY
KEY ("EMPID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
--------------------------------------------------------
--  Constraints for Table ACCOUNT
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."ACCOUNT" MODIFY ("ACCOUNTID" NOT NULL ENABLE);
```

```
  ALTER TABLE "DBST_USER"."ACCOUNT" MODIFY ("ACCOUNTINITIALBALANCE" NOT NULL
ENABLE);
  ALTER TABLE "DBST_USER"."ACCOUNT" MODIFY ("ACCOUNTCURRENTBALANCE" NOT NULL
ENABLE);
  ALTER TABLE "DBST_USER"."ACCOUNT" MODIFY ("ACCOUNTSTATUS" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."ACCOUNT" MODIFY ("MONTHLYPAYMENT" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."ACCOUNT" MODIFY ("BRANCH_BRANCHID" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."ACCOUNT" MODIFY ("BANKEMPLOYEE_EMPID" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."ACCOUNT" ADD CONSTRAINT "ACCOUNT_PK" PRIMARY KEY
("ACCOUNTID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
--------------------------------------------------------
--  Constraints for Table TRANSACTION
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."TRANSACTION" MODIFY ("TRANSACTIONID" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."TRANSACTION" MODIFY ("AMOUNT" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."TRANSACTION" MODIFY ("DATEOFTRANSACTION" NOT NULL
ENABLE);
  ALTER TABLE "DBST_USER"."TRANSACTION" MODIFY ("TRANSACTIONTYPE" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."TRANSACTION" MODIFY ("ACCOUNT_ACCOUNTID" NOT NULL
ENABLE);
  ALTER TABLE "DBST_USER"."TRANSACTION" ADD CONSTRAINT "TRANSACTION_PK" PRIMARY KEY
("TRANSACTIONID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
--------------------------------------------------------
--  Constraints for Table STUDENT
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."STUDENT" MODIFY ("STUDENTID" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."STUDENT" MODIFY ("STUDENTADDRESS_STUDENTADDRESSID" NOT
NULL ENABLE);
  ALTER TABLE "DBST_USER"."STUDENT" MODIFY ("ACCOUNT_ACCOUNTID" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."STUDENT" ADD CONSTRAINT "STUDENT_PK" PRIMARY KEY
("STUDENTID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
```

```
  TABLESPACE "USERS"  ENABLE;
--------------------------------------------------------
--  Constraints for Table LOAN
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."LOAN" MODIFY ("LOANID" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."LOAN" MODIFY ("LOANNAME" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."LOAN" MODIFY ("DATE_ISSUED" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."LOAN" MODIFY ("BALANCE" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."LOAN" MODIFY ("STUDENT_STUDENTID" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."LOAN" MODIFY ("ACCOUNT_ACCOUNTID" NOT NULL ENABLE);
  ALTER TABLE "DBST_USER"."LOAN" ADD CONSTRAINT "LOAN_PK" PRIMARY KEY ("LOANID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
--------------------------------------------------------
--  Ref Constraints for Table STUDENTADDRESS
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."STUDENTADDRESS" ADD CONSTRAINT
"STUDENTADDRESS_ZIPCODE_FK" FOREIGN KEY ("ZIPCODE_ZIPCODE")
        REFERENCES "DBST_USER"."ZIPCODE" ("ZIPCODE") ENABLE;
--------------------------------------------------------
--  Ref Constraints for Table BRANCH
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."BRANCH" ADD CONSTRAINT "BRANCH_ZIPCODE_FK" FOREIGN KEY
("ZIPCODE_ZIPCODE")
        REFERENCES "DBST_USER"."ZIPCODE" ("ZIPCODE") ENABLE;
--------------------------------------------------------
--  Ref Constraints for Table BANKEMPLOYEE
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."BANKEMPLOYEE" ADD CONSTRAINT "BANKEMPLOYEE_BRANCH_FK"
FOREIGN KEY ("BRANCH_BRANCHID")
        REFERENCES "DBST_USER"."BRANCH" ("BRANCHID") ENABLE;
--------------------------------------------------------
--  Ref Constraints for Table ACCOUNT
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."ACCOUNT" ADD CONSTRAINT "ACCOUNT_BANKEMPLOYEE_FK"
FOREIGN KEY ("BANKEMPLOYEE_EMPID")
        REFERENCES "DBST_USER"."BANKEMPLOYEE" ("EMPID") ENABLE;
  ALTER TABLE "DBST_USER"."ACCOUNT" ADD CONSTRAINT "ACCOUNT_BRANCH_FK" FOREIGN KEY
("BRANCH_BRANCHID")
```

```
        REFERENCES "DBST_USER"."BRANCH" ("BRANCHID") ENABLE;
--------------------------------------------------------
--  Ref Constraints for Table TRANSACTION
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."TRANSACTION" ADD CONSTRAINT "TRANSACTION_ACCOUNT_FK"
FOREIGN KEY ("ACCOUNT_ACCOUNTID")
        REFERENCES "DBST_USER"."ACCOUNT" ("ACCOUNTID") ENABLE;
--------------------------------------------------------
--  Ref Constraints for Table STUDENT
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."STUDENT" ADD CONSTRAINT "STUDENT_ACCOUNT_FK" FOREIGN KEY
("ACCOUNT_ACCOUNTID")
        REFERENCES "DBST_USER"."ACCOUNT" ("ACCOUNTID") ENABLE;
  ALTER TABLE "DBST_USER"."STUDENT" ADD CONSTRAINT "STUDENT_STUDENTADDRESS_FK"
FOREIGN KEY ("STUDENTADDRESS_STUDENTADDRESSID")
        REFERENCES "DBST_USER"."STUDENTADDRESS" ("STUDENTADDRESSID") ENABLE;
--------------------------------------------------------
--  Ref Constraints for Table LOAN
--------------------------------------------------------

  ALTER TABLE "DBST_USER"."LOAN" ADD CONSTRAINT "LOAN_ACCOUNT_FK" FOREIGN KEY
("ACCOUNT_ACCOUNTID")
        REFERENCES "DBST_USER"."ACCOUNT" ("ACCOUNTID") ENABLE;
  ALTER TABLE "DBST_USER"."LOAN" ADD CONSTRAINT "LOAN_STUDENT_FK" FOREIGN KEY
("STUDENT_STUDENTID")
        REFERENCES "DBST_USER"."STUDENT" ("STUDENTID") ENABLE;
COMMIT;
```

Fragments and Views

Using the following code. The table is fragmented as such:

```
----------FRAGMENTATION----------
INSERT INTO STUDENTADDRESS@bank_a_b (
    STUDENTADDRESSID,
    STUDENTADDRESS,
       ZIPCODE_ZIPCODE
    )
SELECT
    studentaddressid STUDENTADDRESSID,
studentaddress STUDENTADDRESS,
zipcode_zipcode ZIPCODE_ZIPCODE
FROM (
```

```
    SELECT
    SA.STUDENTADDRESSID,
    SA.STUDENTADDRESS,
       SA.ZIPCODE_ZIPCODE
    FROM STUDENTADDRESS SA
    JOIN ZIPCODE Z
        ON SA.ZIPCODE_ZIPCODE = Z.ZIPCODE
    WHERE Z.STATE = 'MD' OR Z.STATE = 'NC'
    )
;

INSERT INTO BRANCH@bank_a_b (
BRANCHID,
BRANCHNAME,
ZIPCODE_ZIPCODE
)
SELECT
BRANCHID BRANCHID,
BRANCHNAME BRANCHNAME,
ZIPCODE_ZIPCODE ZIPCODE_ZIPCODE
FROM (
SELECT
B.BRANCHID,
B.BRANCHNAME,
B.ZIPCODE_ZIPCODE
FROM BRANCH B
JOIN ZIPCODE Z
ON B.ZIPCODE_ZIPCODE = Z.ZIPCODE
WHERE Z.STATE = 'MD' OR Z.STATE = 'NC'
)
;
COMMIT;

INSERT INTO ZIPCODE@bank_a_b (
    ZIPCODE, CITY,
    STATE, COUNTRY
    )
SELECT
    *
FROM (
    SELECT
        ZIPCODE, CITY,
        STATE, COUNTRY
    FROM
        ZIPCODE
    WHERE
        STATE = 'MD' OR STATE = 'NC'
```

```
    ORDER BY
        ZIPCODE
    )
;
COMMIT;
```

The following is proof of fragmentation:

Figure 5

Proof of Fragmentation running successfully

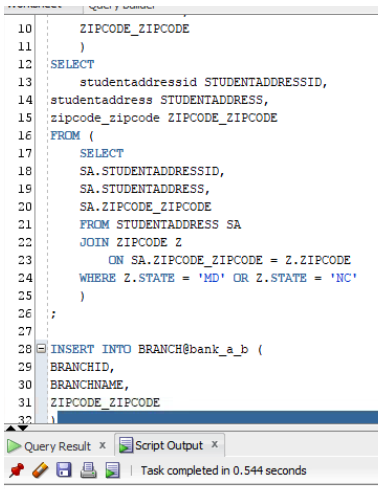
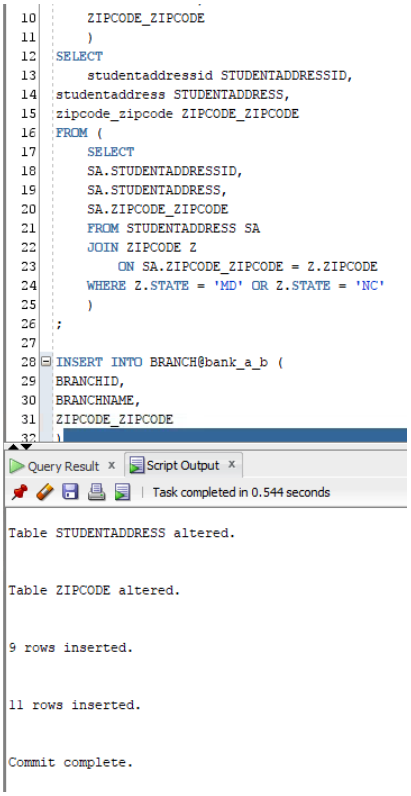


Figure 6

Proof of Fragmentation running successfully



For quick data fetching, five views are created. A view to fetch all BRANCH, STUDENTADDRESS, STUDENT, ZIPCODE, and EMPLOYEE data across both databases is created. The views will be run on DBST663a and shown in the screenshots. Using the following queries:

```
---------To fetch the branches---------
CREATE VIEW ALL_BRANCH_VIEW AS
SELECT
	*
FROM
	Branch
UNION
SELECT
	*
FROM
	BRANCH@bank_a_b;
SELECT * FROM ALL_BRANCH_VIEW;
---------To view all loans distributed to students--------
CREATE VIEW ALL_STUDENTADDRESS_VIEW AS
SELECT
```

```
	*
FROM
	STUDENTADDRESS
UNION
SELECT
	*
FROM
	STUDENTADDRESS@bank_a_b;
SELECT * FROM ALL_STUDENTADDRESS_VIEW;
---------View all students and their respective branch------
CREATE VIEW ALL_STUDENT_VIEW AS
SELECT
	S.STUDENTID,
	S.LASTNAME,
	B.BRANCHNAME,
	Z.ZIPCODE
FROM
	STUDENT S
	JOIN STUDENTADDRESS SA
		ON SA.STUDENTADDRESSID = S.STUDENTADDRESS_STUDENTADDRESSID
	JOIN BRANCH B
		ON B.ZIPCODE_ZIPCODE = SA.ZIPCODE_ZIPCODE
	JOIN ZIPCODE Z
		ON Z.ZIPCODE = B.ZIPCODE_ZIPCODE
UNION
SELECT
	S.STUDENTID,
	S.LASTNAME,
	B.BRANCHNAME,
	Z.ZIPCODE
FROM
	STUDENT@bank_a_b S
	JOIN STUDENTADDRESS@bank_a_b SA
		ON SA.STUDENTADDRESSID = S.STUDENTADDRESS_STUDENTADDRESSID
	JOIN BRANCH@bank_a_b B
		ON B.ZIPCODE_ZIPCODE = SA.ZIPCODE_ZIPCODE
	JOIN ZIPCODE@bank_a_b Z
		ON Z.ZIPCODE = B.ZIPCODE_ZIPCODE;
SELECT * FROM ALL_STUDENTADDRESS_VIEW;
---------See account transactions---------
CREATE VIEW ALL_ZIPCODE_VIEW AS
SELECT
	*
FROM
	ZIPCODE
UNION
SELECT
```

```
	*
FROM
	ZIPCODE@bank_a_b;
SELECT * FROM ALL_ZIPCODE_VIEW;
---------Fetch the Bank Employees and their respective Branch----
CREATE VIEW ALL_EMPLOYEES_VIEW AS
SELECT
	E.EMPID,
	E.LAST_NAME,
	E.ROLE,
	E.PHONE,
	E.EMAIL,
	B.BRANCHID
	B.BRANCHNAME
FROM
	BRANCH B
	JOIN BANKEMPLOYEE E
		ON B.BRANCHID = E.BRANCH_BRANCHID
UNION
SELECT
	E.EMPID,
	E.LAST_NAME,
	E.ROLE,
	E.PHONE,
	E.EMAIL,
	B.BRANCHID,
	B.BRANCHNAME
FROM
	BRANCH@bank_a_b B
	JOIN BANKEMPLOYEE@bank_a_b E
		ON B.BRANCHID = E.BRANCH_BRANCHID;
SELECT * FROM ALL_EMPLOYEES_VIEW;
COMMIT;
```

The following are screenshots proving that the queries work:

Figure 7

BRANCH view creation

```
SELECT * FROM ALL_BRANCH_VIEW;
```

Query Result | SQL | All Rows Fetched: 15 in 0.018 seconds

	BRANCHID	BRANCHNAME	ZIPCODE_ZIPCODE
1	16	Climb Bank Charlotte	28262
2	17	Climb Bank Alabama	36104
3	18	Well Fargo Maryland	20588
4	19	Climb Bank Arizona	85001
5	20	Bank of America NC-C	28262
6	21	Climb Bank Waldorf	20588
7	22	Fith Third Bank NC-C	28213
8	23	Climb Bank Waldorf	20588
9	24	Climb Bank Maryland	20874
10	25	Climb Bank Waldorf North	20588
11	26	Climb Bank Waldorf South	20588
12	27	Well fargo Raleigh North	27601
13	28	Bank of America Indiana	46225
14	29	Well Fargo Indiana	46225
15	30	Well Fargo Charlotte	28262

Figure 8

STUDENTADDRESS view creation

```
CREATE VIEW ALL_STUDENTADDRESS_VIEW AS
SELECT
    *
FROM
    STUDENTADDRESS
UNION
SELECT
    *
FROM
    STUDENTADDRESS@bank_a_b;
```

Script Output | Query Result | Task completed in 0.126 seconds

```
View ALL_STUDENTADDRESS_VIEW created.
```

Figure 9

STUDENT view creation

```
CREATE VIEW ALL_STUDENT_VIEW AS
SELECT
    S.STUDENTID,
    S.LASTNAME,
    B.BRANCHNAME,
    Z.ZIPCODE
FROM
    STUDENT S
    JOIN STUDENTADDRESS SA
        ON SA.STUDENTADDRESSID = S.STUDENTADDRESS_STUDENTADDRESSID
    JOIN BRANCH B
        ON B.ZIPCODE_ZIPCODE = SA.ZIPCODE_ZIPCODE
    JOIN ZIPCODE Z
        ON Z.ZIPCODE = B.ZIPCODE_ZIPCODE
UNION
SELECT
    S.STUDENTID,
    S.LASTNAME,
    B.BRANCHNAME,
    Z.ZIPCODE
FROM
    STUDENT@bank_a_b S
    JOIN STUDENTADDRESS@bank_a_b SA
        ON SA.STUDENTADDRESSID = S.STUDENTADDRESS_STUDENTADDRESSID
    JOIN BRANCH@bank_a_b B
        ON B.ZIPCODE_ZIPCODE = SA.ZIPCODE_ZIPCODE
    JOIN ZIPCODE@bank_a_b Z
        ON Z.ZIPCODE = B.ZIPCODE_ZIPCODE;
SELECT * FROM ALL_STUDENT_VIEW;
```

Script Output | Query Result | Task completed in 0.126 seconds

```
View ALL_STUDENTADDRESS_VIEW created.
```

Figure 10

ZIPCODE view creation

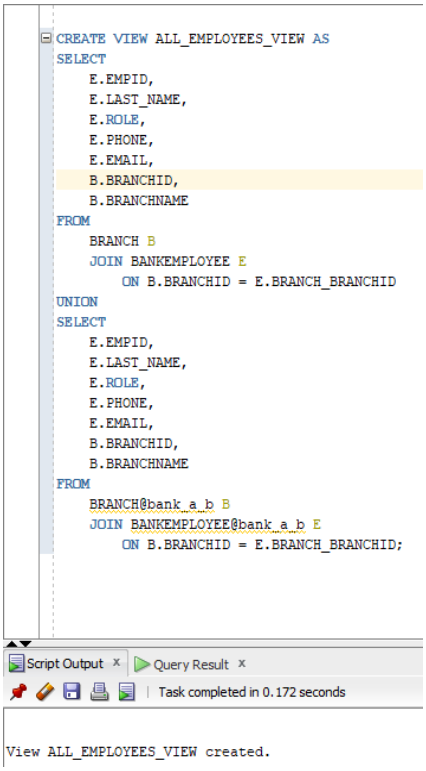
```
CREATE VIEW ALL_ZIPCODE_VIEW AS
SELECT
    *
FROM
    ZIPCODE
UNION
SELECT
    *
FROM
    ZIPCODE@bank_a_b;
SELECT * FROM ALL_ZIPCODE_VIEW;
```

Script Output | Query Result | Task completed in 0.114 seconds

```
View ALL_ZIPCODE_VIEW created.
```

Figure 11

EMPLOYEE view creation



Database Administration

Performance Monitoring

Performance tests are done using the EXPLAIN and TIMING queries. The tests result in quick fetching of the table views considering the JOINs used. The longest was 0.04 seconds and the quickest was under 0.01 seconds. First, the explain queries are shown then the timing queries.

Figure 12

BRANCH EXPLAIN and TIMING queries

```
SELECT PLAN_TABLE_OUTPUT
FROM
    TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE',NULL,'basic'));
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 12 in 0.1 seconds

```
PLAN_TABLE_OUTPUT
Plan hash value: 1049017342

-------------------------------------------------
| Id  | Operation             | Name             |
-------------------------------------------------
|   0 | SELECT STATEMENT      |                  |
|   1 |  VIEW                 | ALL_BRANCH_VIEW  |
|   2 |   SORT UNIQUE         |                  |
|   3 |    UNION-ALL          |                  |
|   4 |     TABLE ACCESS FULL | BRANCH           |
|   5 |     REMOTE            | BRANCH           |
-------------------------------------------------
```

```
SELECT * FROM ALL_BRANCH_VIEW;
```

Script Output | Query Result | Query Result 1

Task completed in 0.023 seconds

```
        19 Climb Bank Arizona       85001
        20 Bank of America NC-C     28262
        21 Climb Bank Waldorf       20588
        22 Fith Third Bank NC-C     28213
        23 Climb Bank Waldorf       20588
        24 Climb Bank Maryland      20874
        25 Climb Bank Waldorf North 20588
        26 Climb Bank Waldorf South 20588

  BRANCHID BRANCHNAME               ZIPCODE_ZI
---------- ------------------------ ----------
        27 Well fargo Raleigh North 27601
        28 Bank of America Indiana  46225
        29 Well Fargo Indiana       46225
        30 Well Fargo Charlotte     28262

15 rows selected.

Elapsed: 00:00:00.006
```

Figure 13

STUDENTADDRESS EXPLAIN and TIMING queries

```
SELECT PLAN_TABLE_OUTPUT
FROM
    TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE',NULL,'basic'));
```

Script Output | Query Result | Query Result 3

SQL | All Rows Fetched: 12 in 0.062 seconds

```
PLAN_TABLE_OUTPUT
Plan hash value: 2498129710

-----------------------------------------------------
| Id  | Operation            | Name                 |
-----------------------------------------------------
|   0 | SELECT STATEMENT     |                      |
|   1 |  VIEW                | ALL_STUDENTADDRESS_VIEW |
|   2 |   SORT UNIQUE        |                      |
|   3 |    UNION-ALL         |                      |
|   4 |     TABLE ACCESS FULL| STUDENTADDRESS       |
|   5 |     REMOTE           | STUDENTADDRESS       |
-----------------------------------------------------
```

```
SELECT * FROM ALL_STUDENTADDRESS_VIEW;
```

Script Output | Query Result | Query Result 1

Task completed in 0.069 seconds

```
19 2321 Mexico Street Road
20 1232 School Street Road
21 1032 Saddle Center
22 98012 Banks Street
23 1106 Featherbrook Road
24 8827 Auburn whisper Ln
25 3557 Sharon Amity Rd
26 9711 David Taylor Dr

STUDENTADDRESSID STUDENTADDRESS
---------------- ------------------------------
              27 3805 Tiffany Rose PI
              28 5300 N Tryon Street
              29 Daisy Moore Ln
              30 9331 JW Clay Blvd

15 rows selected.

Elapsed: 00:00:00.011
```

Figure 14

STUDENT EXPLAIN and TIMING queries

```
SELECT PLAN_TABLE_OUTPUT
FROM
    TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE',NULL,'basic'));
```

Script Output x | Query Result x | Query Result 1 x

SQL | All Rows Fetched: 18 in 0.051 seconds

```
PLAN_TABLE_OUTPUT
Plan hash value: 1016637197

-----------------------------------------------------------------
| Id  | Operation                      | Name              |
-----------------------------------------------------------------
|   0 | SELECT STATEMENT               |                   |
|   1 |  VIEW                          | ALL_STUDENT_VIEW  |
|   2 |   SORT UNIQUE                  |                   |
|   3 |    UNION-ALL                   |                   |
|   4 |     HASH JOIN                  |                   |
|   5 |      MERGE JOIN                |                   |
|   6 |       TABLE ACCESS BY INDEX ROWID| STUDENT         |
|   7 |        INDEX FULL SCAN         | STUDENT__IDX      |
|   8 |       SORT JOIN                |                   |
|   9 |        TABLE ACCESS FULL       | STUDENTADDRESS    |
|  10 |      TABLE ACCESS FULL         | BRANCH            |
|  11 |     REMOTE                     |                   |
-----------------------------------------------------------------
```

```
SELECT * FROM ALL_STUDENT_VIEW;
```

Script Output | Query Result | Query Result 1

Task completed in 0.031 seconds

```
       216 Romaric                  Climb Bank Maryland       20874
       217 Tozan                    Climb Bank Arizona        85001
       218 Smith                    Climb Bank Alabama        36104
       219 Smith                    Fith Third Bank NC-C      28213
       220 Sally                    Climb Bank Alabama        36104
       221 Sally                    Climb Bank Arizona        85001
       222 Sally                    Well fargo Raleigh North  27601
       223 Torres                   Bank of America Indiana   46225
       223 Torres                   Well Fargo Indiana        46225

 STUDENTID LASTNAME                 BRANCHNAME                ZIPCODE
---------- ------------------------ ------------------------- --------
       224 John                     Bank of America NC-C      28262
       224 John                     Climb Bank Charlotte      28262
       224 John                     Well Fargo Charlotte      28262

25 rows selected.

Elapsed: 00:00:00.011
```

Figure 15

ZIPCODE EXPLAIN and TIMING queries

```
SELECT PLAN_TABLE_OUTPUT
FROM
    TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE',NULL,'basic'));
```

Script Output | Query Result

SQL | All Rows Fetched: 12 in 0.12 seconds

```
PLAN_TABLE_OUTPUT
Plan hash value: 3907641964

-----------------------------------------------
| Id  | Operation            | Name            |
-----------------------------------------------
|   0 | SELECT STATEMENT     |                 |
|   1 |  VIEW                | ALL_ZIPCODE_VIEW |
|   2 |   SORT UNIQUE        |                 |
|   3 |    UNION-ALL         |                 |
|   4 |     TABLE ACCESS FULL| ZIPCODE         |
|   5 |     REMOTE           | ZIPCODE         |
-----------------------------------------------
```

```
SET TIMING ON;
SELECT * FROM ALL_ZIPCODE_VIEW;
```

Script Output | Query Result

Task completed in 0.048 seconds

```
ZIPCODE    CITY                      ST COUNTRY
---------- ------------------------- -- -------------
20588      Waldorf                   MD United States
20874      Germantown                MD United States
27601      Raleigh                   NC United States
28213      charlotte                 NC United States
28262      charlotte                 NC United States
36104      Montgomery                AL United States
46225      Indianapolis              IN United States
85001      Phoenix                   AZ United States

8 rows selected.

Elapsed: 00:00:00.028
```

Figure 16

EMPLOYEE EXPLAIN and TIMING queries

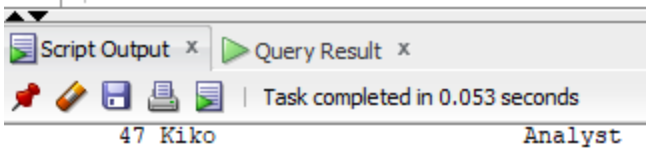
```
SELECT PLAN_TABLE_OUTPUT
FROM
    TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE',NULL,'basic'));
```

Script Output | Query Result

SQL | All Rows Fetched: 16 in 0.044 seconds

```
PLAN_TABLE_OUTPUT
1  Plan hash value: 714932484
2
3  ------------------------------------------------------------------
4  | Id  | Operation                          | Name                |
5  ------------------------------------------------------------------
6  |   0 | SELECT STATEMENT                   |                     |
7  |   1 |  VIEW                              | ALL_EMPLOYEES_VIEW  |
8  |   2 |   SORT UNIQUE                      |                     |
9  |   3 |    UNION-ALL                       |                     |
10 |   4 |     MERGE JOIN                     |                     |
11 |   5 |      TABLE ACCESS BY INDEX ROWID   | BRANCH              |
12 |   6 |       INDEX FULL SCAN              | BRANCH_PK           |
13 |   7 |      SORT JOIN                     |                     |
14 |   8 |       TABLE ACCESS FULL            | BANKEMPLOYEE        |
15 |   9 |     REMOTE                         |                     |
16 ------------------------------------------------------------------
```

```
SELECT * FROM ALL_EMPLOYEES_VIEW;
```




```
        47 Kiko                           Analyst
        48 Morgan                         Manager
        49 Claude                         Teller
        50 Yolonda                        Commercia
        51 Bia                            Loan Anal
        52 Kumaga                         Teller

     EMPID LAST_NAME                      ROLE
---------- ------------------------- ---------
        53 Akoukum                        Loan Anal
        54 Blinde                         IT Analys
        55 Badou                          Analyst
        56 Smith                          Developer
        57 kungo                          Loan Anal
        58 Atta                           Loan Anal
        59 Badou                          Financal
        60 jessica                        Teller

30 rows selected.

Elapsed: 00:00:00.040
```

References

Cherry Road Consultancy Team. (2021, November 1). What are the top 8 reasons to use an Oracle database? CherryRoad Technologies. https://www.cherryroad.com/2021/10/22/oracle-database-cloud/

Tutorials Point. (2021, July 3). Construct an ER diagram for the banking system in DBMS? Biggest Online Tutorials Library. https://www.tutorialspoint.com/construct-an-er-diagram-for-the-banking-system-in-dbms