# Banking Management System Database

## Designed, developed, and implemented by Xinheng Lu

### Project Objectives

The purpose of this project is to design, develop, and implement a Database Tier using Oracle 11g and Oracle SQL Developer for an application named *EZBanking.com* in order to manage customers, bank accounts, transactions, loans, etc. in different banking locations around the world.

### Planning Phase – Business Requirements

NYCTech Bank global bank with $14.1 billion in assets & $2.3 billion in market capital. NYCTech Bank operates 600 branches throughout the US and another 300 internationally in addition to 7000 ATM machines in branches and other public and business locations. With branch in US, Canada, Mexico, UK, Japan & Australia. For example, New York City has two branches in Manhattan, one in Brooklyn and two in Queens. It has 6800 employees and is the 40th largest bank in the US. It is one of the few banks that can credibly describe themselves as conservative with consistent shareholder retention.

Customers make banking transactions on their bank accounts at banking locations throughout the world. Transactions include getting their account balance, making deposits & withdraws from their accounts, in addition to transferring money from one account to another. There are only two types of banking locations, bank branches and ATM machines. A bank branch is identified by a banking location id, banking location address which is composed of number-street, city, state, zip code & country. In addition, a unique branch number, branch name, the branch manager's name composed of manager first name & manager last name, branch phone number and email address. For example, we have a branch in Brooklyn USA whose banking Location id is 7844, located at 323 Flatbush Avenue, Brooklyn New York, 11223, USA branch number is 1010 and name Flatbush NYCTech Bank, managed by Carol Mathews, phone number 1-718-823-2100 and email *Flatbush-branch@nyctechbank.com*.

An ATM location is identified also by a banking location id, banking location address which is composed of number-street, city, state, zip code & country. In addition, a unique ATM Number an ATM cash limit. A banking location such as a branch may or may not have an ATM on premise, or a banking location may just be an ATM residing offsite from a branch, situated in other places such as shopping centers, airports, railways stations and other retail locations. Therefore, these banking locations can reside together or separate from each other. A banking location is either a branch or ATM, there are no other types of banking locations we have.

NYCTech Bank offers its banking services to primarily two types of customers, corporate/commercial & an individual person. A customer is identified by a customer ID, customer address which is composed of number-street & name, city, state, zip code and country. In addition, email address, phone number, person's name composed of (first & last) social security number, date of birth, mobile number, person age, gender and beneficiary in case of death. A consumer/person customer can have more than one beneficiary. Each beneficiary must have beneficiary's social security number, first name, last name and date of birth. Note that a beneficiary can be beneficiary to more than one bank customer. In addition, a customer has a company name if the customer is a corporate customer, their unique company federal tax id and finally company primary contact composed of contact name, contact phone number & email. A customer must be either a person, corporation or other, therefore there are other types of customers we need to track other than a person or corporation such as governments etc., but for now such customer types is out of scope.

Customers usually have two types of what we call simple bank accounts, which are checking and savings. For a checking account, we need to store the account number, present balance, available balance, account status ID or ID of the status which is described and stored in account status description with values: opened, closed, suspended, etc. In addition, service charge, overdraft protection status (activated, deactivated) and overdraft protection limit. For a savings account we need to capture the account number, present balance, available balance, account status ID or ID of the status which is described and stored in account status description with values: opened, closed, suspended, etc. In addition, interest rate and excessive withdraw penalty charge, which is a charge when a customer withdraws more than 10 times in one month. Note that there are other types of accounts other than checking or savings that need to be tracked such as credit card accounts etc. Details on these types of accounts will be added in future versions of the application, for now we just need to store them as accounts storing only their account number, available & present balances etc.

A customer makes a banking transaction to an account at a banking location. A customer can make any number of transactions to any banking locations (branch or ATM). Likewise, a banking location can receive many transactions from different customers. A transaction includes getting checking or savings account balances, deposits to checking or savings accounts, transfers to and from checking and savings accounts. When a customer makes a banking transaction at a banking location, we need to capture a transaction id, transaction type ID or ID number assigned to the transaction executed and described and stored in transaction type description whose values are as follows: Checking-Balance, Checking-Deposit, Checking-Withdraw, Savings-Balance, Savings-Deposit, Savings-Withdraw, Transfer-Savings-to-Checking, Transfer-Checking-to-Savings. In addition, we need to capture transaction amount, date of transaction and time of transaction.

To be a customer, a customer must create/open a Bank Account at a local Branch, which becomes the customer primary branch. When the account is opened, we need to capture the date the account was opened, date account is closed and the type of account created (Checking, Savings or other).

Corporate customers must be assigned a personal banker. This will be one bank employee that will address and provide customer service to the corporate customer. The personal banker is identified by a banker id, banker name (first & last), banker contact information (phone number & email). Note that personal bankers are not dedicated to one customer but must have at least one corporate customer to be a personal banker.

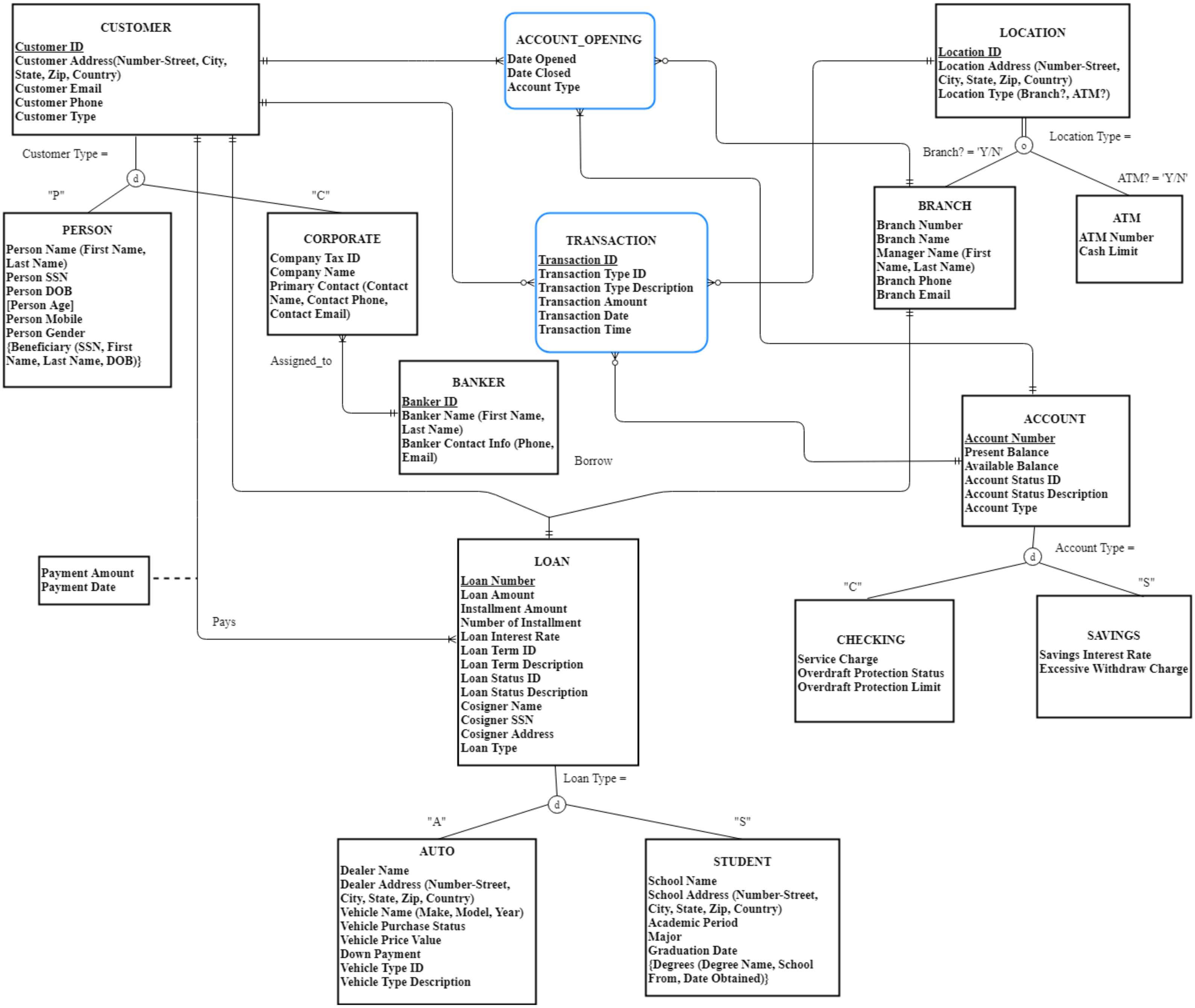
Our bank also provides loan services such as student, auto and other types of loans. A student loan is identified by a loan number, loan amount, monthly installment amount, number of installments, interest rate, loan term ID or ID number assigned and stored in loan term description whose values are: 1 year, 3 years, 5 years, 10 years etc. In addition, loan status ID or number assigned and stored in the attribute loan status description with values such as: approved, in progress, defaulted, delinquent, paid in full, refinanced, death, etc. Other attributes include co-signers name, co-signers social security number and co-signer address, school name, school address composed of number-street & name, city, state, zip code and country. In addition to academic period of loan (Fall 2016 through Spring 2017, etc.), major, estimated graduation date and finally academic degrees student has obtained. Note that students can have many academic degrees and for each degree we need to store the degree, school degree is from, and date degree was obtained.

For an auto loan, we store the loan number which identifies the loan, total loan amount, installment amount, number of installments, interest rate, loan term ID or ID number assigned and stored in loan term description whose values are: 1 year, 3 years, 5 years, 10 years etc. In addition, loan status ID or number assigned and stored in the attribute loan status description with values such as: in progress, defaulted, delinquent, paid in full, refinanced, death, etc. Other attributes include co-signers name, co-signers social security number and co-signer address, auto dealer name, dealer address composed of number-street & name, city, state, zip code and country, vehicle name composed of make, model and year. In addition to vehicle purchase status (new, pre-owned), vehicle full price value, down payment, vehicle type ID which is a number assigned to a vehicle type that is stored in the attribute vehicle type description with values such as passenger car, commercial vehicle, SUV, Truck, etc.

To borrow a loan, a customer must borrow a loan at a local branch. That branch will manage the loan going forward. Customers also make payments to a loan. To make payments the customer must have a loan and can make payments to one or more loans they have, nevertheless, a loan can only belong to one customer. When a payment is made, we need to capture the payment amount and date of payment.

## Analysis Phase – EER Conceptual Model Diagram

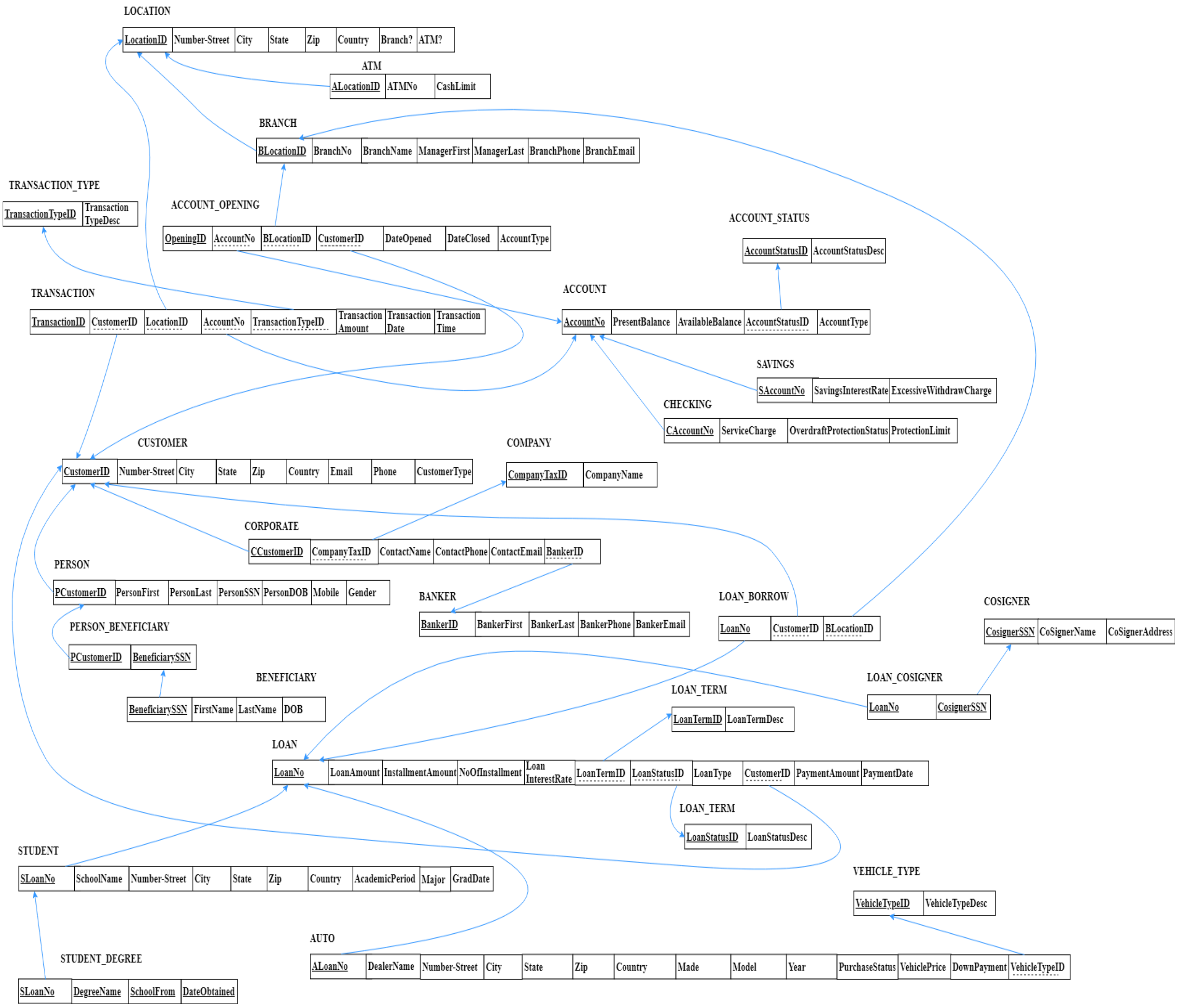
Taken from the business requirements, this phase will create different entities along with their attributes, capture relationships between those entities, and analyze the number of occurrence that is allowed in each entity. The deliverable is shown below.



- Assumed that the cardinality between BRANCH and ACCOUNT_OPENING is optional many because a branch does not have to participant in an account opening event to be a branch

## Design Phase – Normalized Logical Model Diagram

This phase will convert and normalize the EER model from the analysis phase to a Normalized Logical Model representing different tables with relational keys and attributes in the future database system. Based on all specifications and requirements, transitive dependencies are found in TRANSACTION, ACCOUNT, CORPORATE, LOAN, and AUTO tables. After executing third normal form, the final version of Normalized Logical Model is shown below.


LOCATION
LocationID | Number-Street | City | State | Zip | Country | Branch? | ATM?
ATM
ALocationID | ATMNo | CashLimit
BRANCH
BLocationID | BranchNo | BranchName | ManagerFirst | ManagerLast | BranchPhone | BranchEmail
TRANSACTION_TYPE
TransactionTypeID | Transaction TypeDesc
ACCOUNT_OPENING
OpeningID | AccountNo | BLocationID | CustomerID | DateOpened | DateClosed | AccountType
ACCOUNT_STATUS
AccountStatusID | AccountStatusDesc
TRANSACTION
TransactionID | CustomerID | LocationID | AccountNo | TransactionTypeID | Transaction Amount | Transaction Date | Transaction Time
ACCOUNT
AccountNo | PresentBalance | AvailableBalance | AccountStatusID | AccountType
SAVINGS
SAccountNo | SavingsInterestRate | ExcessiveWithdrawCharge
CHECKING
CAccountNo | ServiceCharge | OverdraftProtectionStatus | ProtectionLimit
CUSTOMER
CustomerID | Number-Street | City | State | Zip | Country | Email | Phone | CustomerType
COMPANY
CompanyTaxID | CompanyName
CORPORATE
CCustomerID | CompanyTaxID | ContactName | ContactPhone | ContactEmail | BankerID
PERSON
PCustomerID | PersonFirst | PersonLast | PersonSSN | PersonDOB | Mobile | Gender
BANKER
BankerID | BankerFirst | BankerLast | BankerPhone | BankerEmail
LOAN_BORROW
LoanNo | CustomerID | BLocationID
COSIGNER
CosignerSSN | CoSignerName | CoSignerAddress
PERSON_BENEFICIARY
PCustomerID | BeneficiarySSN
BENEFICIARY
BeneficiarySSN | FirstName | LastName | DOB
LOAN_COSIGNER
LoanNo | CosignerSSN
LOAN_TERM
LoanTermID | LoanTermDesc
LOAN
LoanNo | LoanAmount | InstallmentAmount | NoOfInstallment | Loan InterestRate | LoanTermID | LoanStatusID | LoanType | CustomerID | PaymentAmount | PaymentDate
LOAN_TERM
LoanStatusID | LoanStatusDesc
STUDENT
SLoanNo | SchoolName | Number-Street | City | State | Zip | Country | AcademicPeriod | Major | GradDate
VEHICLE_TYPE
VehicleTypeID | VehicleTypeDesc
STUDENT_DEGREE
SLoanNo | DegreeName | SchoolFrom | DateObtained
AUTO
ALoanNo | DealerName | Number-Street | City | State | Zip | Country | Made | Model | Year | PurchaseStatus | VehiclePrice | DownPayment | VehicleTypeID

- Assumed that person’s date of birth can represent person’s age, thus, person age is opted out in PERSON table
- Assumed that an account opening is identified by an opening ID
- Assumed that a loan can have any number of cosigner and a cosigner can cosigner more than one loan

## Design Phase – Data Dictionary

In this phase, a Data Dictionary will be created containing information such as datatype and constraints about attributes of all the Normalized Logical Models created in previous step.

| LOCATION | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **LocationID** | Number | NUMBER (X) | 4 | Primary Key | Unique identifier for a location instance |
| **Number-Street** | Variable Character | VARCHAR2 (X) | 100 | NOT NULL | Location address number and street name |
| **City** | Variable Character | VARCHAR2 (X) | 75 | NOT NULL | City/Town name |
| **State** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | State/Province name |
| **Zip** | Variable Character | VARCHAR2 (X) | 10 | NOT NULL | Zip/Postal code |
| **Country** | Variable Character | VARCHAR2 (X) | 75 | NOT NULL | Country name |
| **Branch?** | Character | CHAR (X) | 1 | NOT NULL | If the location is a branch (Y/N) |
| **ATM?** | Character | CHAR (X) | 1 | NOT NULL | If the location is an ATM (Y/N) |

- Using VARCHAR2 for State is because country like Mexico and Japan has more than two characters for their states
- Using VARCHAR2 for Zip is because postal code in Canada or other countries is comprised of characters and numbers

| BRANCH | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **BLocationID** | Number | NUMBER (X) | 4 | Primary Key | Unique identifier for a branch instance (Foreign Key of LOCATION table) |
| **BranchNo** | Number | NUMBER (X) | 4 | NOT NULL, UNIQUE | Unique number for a branch instance |
| **BranchName** | Variable Character | VARCHAR2 (X) | 70 | NOT NULL | Branch name |
| **ManagerFirst** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | Manager first name |
| **ManagerLast** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | Manager last name |
| **BranchPhone** | Variable Character | VARCHAR2 (X) | 20 | NOT NULL | Branch phone number |
| **BranchEmail** | Variable Character | VARCHAR2 (X) | 70 | NOT NULL | Branch email |

| ATM | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **ALocationID** | Number | NUMBER (X) | 4 | Primary Key | Unique identifier for an ATM instance (Foreign Key of LOCATION table) |
| **ATMNo** | Number | NUMBER (X) | 4 | NOT NULL, UNIQUE | Unique number for an ATM instance |
| **CashLimit** | Number | NUMBER (X) | 4 | NOT NULL | Maximum withdrawal amount for an ATM |

| CUSTOMER | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **CustomerID** | Number | NUMBER (X) | 5 | Primary Key | Unique identifier for a customer instance |
| **Number-Street** | Variable Character | VARCHAR2 (X) | 100 | NOT NULL | Customer address number and street name |
| **City** | Variable Character | VARCHAR2 (X) | 75 | NOT NULL | City/Town name |
| **State** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | State/Province name |
| **Zip** | Variable Character | VARCHAR2 (X) | 10 | NOT NULL | Zip/Postal code |
| **Country** | Variable Character | VARCHAR2 (X) | 75 | NOT NULL | Country name |
| **Email** | Variable Character | VARCHAR2 (X) | 100 | NOT NULL | Customer email |
| **Phone** | Variable Character | VARCHAR2 (X) | 20 | NOT NULL | Customer phone number |
| **CustomerType** | Character | CHAR (X) | 1 | NOT NULL | Customer Type |

| PERSON | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **PCustomerID** | Number | NUMBER (X) | 5 | Primary Key | Unique identifier for a person instance (Foreign Key of CUSTOMER table) |
| **PersonFirst** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | Person first name |
| **PersonLast** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | Person last name |
| **PersonSSN** | Variable Character | VARCHAR2 (X) | 20 | NOT NULL, UNIQUE | Person Social Security Number |
| **PersonDOB** | Date | DATE | MM/DD/YYYY | NOT NULL | Person Date of Birth |
| **Mobile** | Variable Character | VARCHAR2 (X) | 20 | NOT NULL | Person mobile number |
| **Gender** | Character | CHAR (X) | 1 | NOT NULL | Person gender (M/F) |

- Using size 20 for PersonSSN is because SSN in other countries is comprised of more than nine characters/digits

| BENEFICIARY | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **BeneficiarySSN** | Variable Character | VARCHAR2 (X) | 20 | Primary Key | Beneficiary social security number |
| **FirstName** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | Beneficiary first name |
| **LastName** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | Beneficiary last name |
| **DOB** | Date | DATE | MM/DD/YYYY | NOT NULL | Beneficiary Date of Birth |

| PERSON_BENEFICIARY | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **PCustomerID** | Number | NUMBER (X) | 5 | Primary Key | Foreign Key of PERSON table |

| | | | | | |
|---|---|---|---|---|---|
| **BeneficiarySSN** | Variable Character | VARCHAR2 (X) | 20 | Primary Key | Foreign Key of BENEFICIARY table |

### CORPORATE

| Attribute Name | Data Type | Oracle Data Type | Length/Size /Format | Constraints | Description |
|---|---|---|---|---|---|
| **CCustomerID** | Number | NUMBER (X) | 5 | Primary Key | Unique identifier for a corporate instance (Foreign Key of CUSTOMER table) |
| **CompanyTaxID** | Variable Character | VARCHAR2 (X) | 20 | Foreign Key<br>NOT NULL | Company federal tax ID (Foreign Key of COMPANY table) |
| **ContactName** | Variable Character | VARCHAR2 (X) | 75 | NOT NULL | Contact name |
| **ContactPhone** | Variable Character | VARCHAR2 (X) | 20 | NOT NULL | Contact phone number |
| **ContactEmail** | Variable Character | VARCHAR2 (X) | 100 | NOT NULL | Contact email |
| **BankerID** | Number | NUMBER (X) | 4 | Foreign Key<br>NOT NULL | ID of banker who is assigned to this corporate customer (Foreign Key of BANKER table) |

### COMPANY

| Attribute Name | Data Type | Oracle Data Type | Length/Size /Format | Constraints | Description |
|---|---|---|---|---|---|
| **CompanyTaxID** | Variable Character | VARCHAR2 (X) | 20 | Primary Key | Unique identifier for a company instance |
| **CompanyName** | Variable Character | VARCHAR2 (X) | 80 | NOT NULL | Company name |

### BANKER

| Attribute Name | Data Type | Oracle Data Type | Length/Size /Format | Constraints | Description |
|---|---|---|---|---|---|
| **BankerID** | Number | NUMBER (X) | 4 | Primary Key | Unique identifier for a banker instance |
| **BankerFirst** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | Banker first name |
| **BankerLast** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | Banker last name |
| **BankerPhone** | Variable Character | VARCHAR2 (X) | 20 | NOT NULL | Banker phone number |
| **BankerEmail** | Variable Character | VARCHAR2 (X) | 70 | NOT NULL | Banker email |

### ACCOUNT

| Attribute Name | Data Type | Oracle Data Type | Length/Size /Format | Constraints | Description |
|---|---|---|---|---|---|
| **AccountNo** | Number | NUMBER (X) | 15 | Primary Key | Unique identifier for an account instance |
| **PresentBalance** | Decimal Number | DEC (X, Y) | (10,2) | NOT NULL | Present Balance |
| **AvailableBalance** | Decimal Number | DEC (X, Y) | (10,2) | NOT NULL | Available Balance |
| **AccountStatusID** | Number | NUMBER (X) | 2 | Foreign Key<br>NOT NULL | ID of account status (Foreign Key of ACCOUNT_STATUS table) |

| | | | | | |
|---|---|---|---|---|---|
| **AccountType** | Character | CHAR (X) | 1 | NOT NULL | Account Type |

| ACCOUNT_STATUS | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **AccountStatusID** | Number | NUMBER (X) | 2 | Primary Key | Unique identifier for an account status instance |
| **AccountStatusDesc** | Variable Character | VARCHAR2 (X) | 15 | NOT NULL | Account status description |

| CHECKING | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **CAccountNo** | Number | NUMBER (X) | 15 | Primary Key | Unique identifier for a checking instance (Foreign Key of ACCOUNT table) |
| **ServiceCharge** | Decimal Number | DEC (X, Y) | (4,2) | NOT NULL | Service charge |
| **OverdraftProtectionStatus** | Variable Character | VARCHAR2 (X) | 11 | NOT NULL | Overdraft protection status (activated, deactivated) |
| **ProtectionLimit** | Decimal Number | DEC (X, Y) | (10,2) | NULL | Overdraft protection limit |

| SAVINGS | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **SAccountNo** | Number | NUMBER (X) | 15 | Primary Key | Unique identifier for a savings instance (Foreign Key of ACCOUNT table) |
| **SavingsInterestRate** | Decimal Number | DEC (X, Y) | (3,2) | NOT NULL | Interest rate |
| **ExcessiveWithdrawCharge** | Decimal Number | DEC (X, Y) | (4,2) | NOT NULL | Excessive withdraw penalty over 10 times in a month |

| ACCOUNT_OPENING | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **OpeningID** | Number | NUMBER (X) | 5 | Primary Key | Unique identifier for an account_opening instance |
| **AccountNo** | Number | NUMBER (X) | 15 | Foreign Key<br>NOT NULL | ID of account that was opened (Foreign Key of ACCOUNT table) |
| **BLocationID** | Number | NUMBER (X) | 4 | Foreign Key<br>NOT NULL | ID of branch that was opened at (Foreign Key of BRANCH table) |

| | | | | | |
|---|---|---|---|---|---|
| **CustomerID** | Number | NUMBER (X) | 5 | Foreign Key<br>NOT NULL | ID of customer who opened (Foreign Key of CUSTOMER table) |
| **DateOpen** | Date | DATE | MM/DD/YYYY | NOT NULL | Date account opened |
| **DateClosed** | Date | DATE | MM/DD/YYYY | NOT NULL | Date account closed |
| **AccountType** | Variable Character | VARCHAR2 (X) | 20 | NOT NULL | Account Type |

| **TRANSACTION** | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **TransactionID** | Number | NUMBER (X) | 8 | Primary Key | Unique identifier for a transaction instance |
| **CustomerID** | Number | NUMBER (X) | 5 | Foreign Key<br>NOT NULL | ID of customer  who performed (Foreign Key of CUSTOMER table) |
| **LocationID** | Number | NUMBER (X) | 4 | Foreign Key<br>NOT NULL | ID of location where was performed at (Foreign Key of LOCATION table) |
| **AccountNo** | Number | NUMBER (X) | 15 | Foreign Key<br>NOT NULL | ID of account that was performed for (Foreign Key of ACCOUNT table) |
| **TransactionTypeID** | Number | NUMBER (X) | 2 | Foreign Key<br>NOT NULL | ID of transaction type (Foreign Key of TRANSACTION_TYPE table) |
| **TransactionAmount** | Decimal Number | DEC (X, Y) | (10,2) | NOT NULL | Transaction amount |
| **TransactionDate** | Date | DATE | MM/DD/YYYY | NOT NULL | Transaction date |
| **TransactionTime** | Number | NUMBER (X) | 4 | NOT NULL | Transaction type |

| **TRANSACTION_TYPE** | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **TransactionTypeID** | Number | NUMBER (X) | 2 | Foreign Key | Unique identifier for a transaction  type instance |
| **TransactionTypeDesc** | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | Transaction type description |

| **LOAN** | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **LoanNo** | Number | NUMBER (X) | 10 | Primary Key | Unique identifier for a loan instance |
| **LoanAmount** | Decimal Number | DEC (X, Y) | (9,2) | NOT NULL | Loan amount |
| **InstallmentAmount** | Decimal Number | DEC (X, Y) | (9,2) | NOT NULL | Monthly payment amount |

| | | | | | |
|---|---|---|---|---|---|
| **NoOfInstallment** | Number | NUMBER (X) | 3 | NOT NULL | Number of months to pay off |
| **InterestRate** | Decimal Number | DEC (X, Y) | (3,2) | NOT NULL | Loan interest rate |
| **LoanTermID** | Number | NUMBER (X) | 2 | Foreign Key<br>NOT NULL | ID of loan term (Foreign Key of LOAN_TERM table) |
| **LoanStatusID** | Number | NUMBER (X) | 2 | Foreign Key<br>NOT NULL | ID of loan status (Foreign Key of LOAN_STATUS table) |
| **LoanType** | Character | CHAR (X) | 1 | NOT NULL | Loan Type |
| **CustomerID** | Number | NUMBER (X) | 5 | Foreign Key<br>NOT NULL | Foreign Key of CUSTOMER table |
| **PaymentAmount** | Decimal Number | DEC (X, Y) | (9,2) | NOT NULL | Payment amount |
| **PaymentDate** | Date | DATE | MM/DD/YYYY | NOT NULL | Payment date |

| **COSIGNER** | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **CosignerSSN** | Variable Character | VARCHAR2 (X) | 20 | Primary Key | Unique identifier for a cosigner instance |
| **CosignerName** | Variable Character | VARCHAR2 (X) | 75 | NOT NULL | Cosigner full name |
| **CosignerAddress** | Variable Character | VARCHAR2 (X) | 150 | NOT NULL | Cosigner full address |

| **LOAN_COSIGNER** | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **LoanNo** | Number | NUMBER (X) | 10 | Primary Key | Foreign Key of LOAN table |
| **CosignerSSN** | Variable Character | VARCHAR2 (X) | 20 | Primary Key | Foreign Key of COSIGNER table |

| **LOAN_TERM** | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **LoanTermID** | Number | NUMBER (X) | 2 | Primary Key | Unique identifier for a loan term instance |
| **LoanTermDesc** | Variable Character | VARCHAR2 (X) | 10 | NOT NULL | Loan term description |

| **LOAN_STATUS** | | | | | |
|---|---|---|---|---|---|
| **Attribute Name** | **Data Type** | **Oracle Data Type** | **Length/Size /Format** | **Constraints** | **Description** |
| **LoanStatusID** | Number | NUMBER (X) | 2 | Primary Key | Unique identifier for a loan status instance |
| **LoanStatusDesc** | Variable Character | VARCHAR2 (X) | 20 | NOT NULL | Loan status description |

**AUTO**

| Attribute Name | Data Type | Oracle Data Type | Length/Size /Format | Constraints | Description |
|---|---|---|---|---|---|
| ALoanNo | Number | NUMBER (X) | 10 | Primary Key | Unique identifier for an auto loan instance (Foreign Key of LOAN table) |
| DealerName | Variable Character | VARCHAR2 (X) | 80 | NOT NULL | Auto dealer name |
| Number-Street | Variable Character | VARCHAR2 (X) | 100 | NOT NULL | dealer address number and street name |
| City | Variable Character | VARCHAR2 (X) | 75 | NOT NULL | City/Town name |
| State | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | State/Province name |
| Zip | Variable Character | VARCHAR2 (X) | 10 | NOT NULL | Zip/Postal code |
| Country | Variable Character | VARCHAR2 (X) | 75 | NOT NULL | Country name |
| Made | Variable Character | VARCHAR2 (X) | 20 | NOT NULL | Vehicle made |
| Model | Variable Character | VARCHAR2 (X) | 20 | NOT NULL | Vehicle model |
| Year | Number | NUMBER (X) | 4 | NOT NULL | Vehicle year |
| PurchaseStatus | Variable Character | VARCHAR2 (X) | 9 | NOT NULL | Purchase status (new, pre-owned) |
| VehiclePrice | Decimal Number | DEC (X, Y) | (9,2) | NOT NULL | Vehicle price value |
| DownPayment | Decimal Number | DEC (X, Y) | (9,2) | NOT NULL | Down payment |
| VehicleTypeID | Number | NUMBER (X) | 2 | Foreign Key<br>NOT NULL | ID of vehicle type (Foreign Key of VEHICLE_TYPE table) |

**VEHICLE_TYPE**

| Attribute Name | Data Type | Oracle Data Type | Length/Size /Format | Constraints | Description |
|---|---|---|---|---|---|
| VehicleTypeID | Number | NUMBER (X) | 2 | Primary Key | Unique identifier for a vehicle_type instance |
| VehicleTypeDesc | Variable Character | VARCHAR2 (X) | 30 | NOT NULL | Vehicle type description |

**STUDENT**

| Attribute Name | Data Type | Oracle Data Type | Length/Size /Format | Constraints | Description |
|---|---|---|---|---|---|
| SLoanNo | Number | NUMBER (X) | 10 | Primary Key | Unique identifier for a student loan instance (Foreign Key of LOAN table) |
| SchoolName | Variable Character | VARCHAR2 (X) | 80 | NOT NULL | School name |
| Number-Street | Variable Character | VARCHAR2 (X) | 100 | NOT NULL | School address number and street name |
| City | Variable Character | VARCHAR2 (X) | 75 | NOT NULL | City/Town name |
| State | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | State/Province name |
| Zip | Variable Character | VARCHAR2 (X) | 10 | NOT NULL | Zip/Postal code |
| Country | Variable Character | VARCHAR2 (X) | 75 | NOT NULL | Country name |
| AcademicPeriod | Variable Character | VARCHAR2 (X) | 50 | NOT NULL | Academic period |

| | | | | | |
|---|---|---|---|---|---|
| **Major** | Variable Character | VARCHAR2 (X) | 80 | NOT NULL | Academic major |
| **GradDate** | Date | DATE | MM/DD/YY | NOT NULL | Estimated graduation date |

**STUDENT_DEGREE**

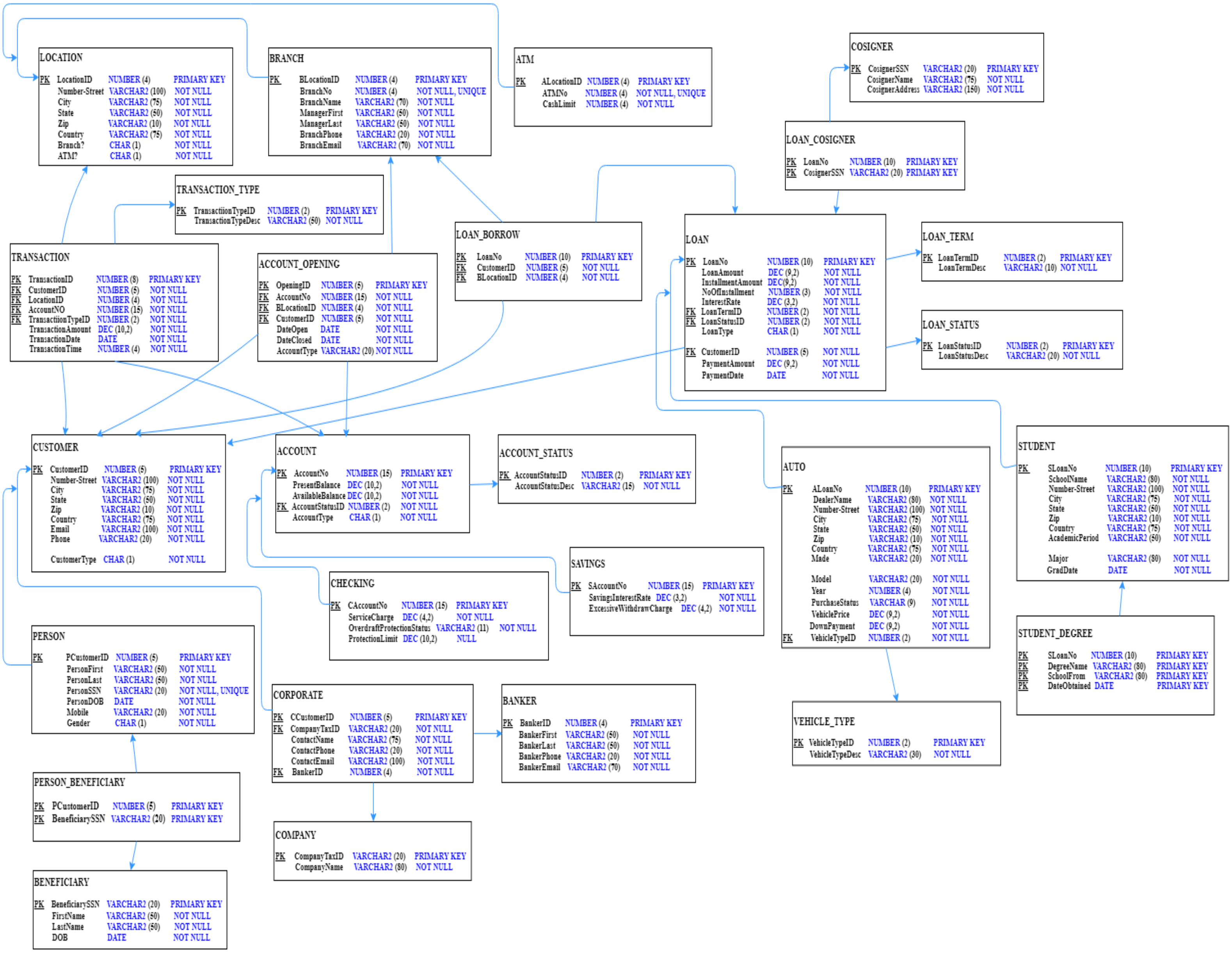
| Attribute Name | Data Type | Oracle Data Type | Length/Size /Format | Constraints | Description |
|---|---|---|---|---|---|
| **SLoanNo** | Number | NUMBER (X) | 10 | Primary Key | Foreign Key of STUDENT table |
| **DegreeName** | Variable Character | VARCHAR2 (X) | 80 | Primary Key | Degree name |
| **SchoolFrom** | Variable Character | VARCHAR2 (X) | 80 | Primary Key | School student graduated from |
| **DateObtained** | Date | DATE | MM/DD/YY | Primary Key | Date student obtained degree |

**LOAN_BORROW**

| Attribute Name | Data Type | Oracle Data Type | Length/Size /Format | Constraints | Description |
|---|---|---|---|---|---|
| **LoanNo** | Number | NUMBER (X) | 10 | Primary Key | Unique identifier for a loan_borrow instance (Foreign Key of LOAN table) |
| **CustomerID** | Number | NUMBER (X) | 5 | Foreign Key<br>NOT NULL | ID of customer who borrowed (Foreign Key of CUSTOMER table) |
| **BLocationID** | Number | NUMBER (X) | 4 | Foreign Key<br>NOT NULL | ID of branch that handled (Foreign Key of BRANCH table) |

## Design Phase – Physical Data Model Diagram

This phase will refer to the Data Dictionary created in previous step and convert the Normalized Logical Model to a Physical Model. The deliverable is shown below.


LOCATION
PK LocationID NUMBER (4) PRIMARY KEY
Number-Street VARCHAR2 (100) NOT NULL
City VARCHAR2 (75) NOT NULL
State VARCHAR2 (50) NOT NULL
Zip VARCHAR2 (10) NOT NULL
Country VARCHAR2 (75) NOT NULL
Branch? CHAR (1) NOT NULL
ATM? CHAR (1) NOT NULL
BRANCH
PK BLocationID NUMBER (4) PRIMARY KEY
BranchNo NUMBER (4) NOT NULL, UNIQUE
BranchName VARCHAR2 (70) NOT NULL
ManagerFirst VARCHAR2 (50) NOT NULL
ManagerLast VARCHAR2 (50) NOT NULL
BranchPhone VARCHAR2 (20) NOT NULL
BranchEmail VARCHAR2 (70) NOT NULL
ATM
PK ALocationID NUMBER (4) PRIMARY KEY
ATMNo NUMBER (4) NOT NULL, UNIQUE
CashLimit NUMBER (4) NOT NULL
COSIGNER
PK CosignerSSN VARCHAR2 (20) PRIMARY KEY
CosignerName VARCHAR2 (75) NOT NULL
CosignerAddress VARCHAR2 (150) NOT NULL
LOAN_COSIGNER
PK LoanNo NUMBER (10) PRIMARY KEY
PK CosignerSSN VARCHAR2 (20) PRIMARY KEY
TRANSACTION_TYPE
PK TransactiionTypeID NUMBER (2) PRIMARY KEY
TransactionTypeDesc VARCHAR2 (50) NOT NULL
LOAN_BORROW
PK LoanNo NUMBER (10) PRIMARY KEY
FK CustomerID NUMBER (5) NOT NULL
FK BLocationID NUMBER (4) NOT NULL
LOAN
PK LoanNo NUMBER (10) PRIMARY KEY
LoanAmount DEC (9,2) NOT NULL
InstallmentAmount DEC(9,2) NOT NULL
NoOfInstallment NUMBER (3) NOT NULL
InterestRate DEC (3,2) NOT NULL
FK LoanTermID NUMBER (2) NOT NULL
FK LoanStatusID NUMBER (2) NOT NULL
LoanType CHAR (1) NOT NULL
FK CustomerID NUMBER (5) NOT NULL
PaymentAmount DEC (9,2) NOT NULL
PaymentDate DATE NOT NULL
LOAN_TERM
PK LoanTermID NUMBER (2) PRIMARY KEY
LoanTermDesc VARCHAR2 (10) NOT NULL
LOAN_STATUS
PK LoanStatusID NUMBER (2) PRIMARY KEY
LoanStatusDesc VARCHAR2 (20) NOT NULL
TRANSACTION
PK TransactionID NUMBER (8) PRIMARY KEY
FK CustomerID NUMBER (5) NOT NULL
FK LocationID NUMBER (4) NOT NULL
FK AccountNO NUMBER (15) NOT NULL
FK TransactiionTypeID NUMBER (2) NOT NULL
TransactionAmount DEC (10,2) NOT NULL
TransactionDate DATE NOT NULL
TransactionTime NUMBER (4) NOT NULL
ACCOUNT_OPENING
PK OpeningID NUMBER (5) PRIMARY KEY
FK AccountNo NUMBER (15) NOT NULL
FK BLocationID NUMBER (4) NOT NULL
FK CustomerID NUMBER (5) NOT NULL
DateOpen DATE NOT NULL
DateClosed DATE NOT NULL
AccountType VARCHAR2 (20) NOT NULL
CUSTOMER
PK CustomerID NUMBER (5) PRIMARY KEY
Number-Street VARCHAR2 (100) NOT NULL
City VARCHAR2 (75) NOT NULL
State VARCHAR2 (50) NOT NULL
Zip VARCHAR2 (10) NOT NULL
Country VARCHAR2 (75) NOT NULL
Email VARCHAR2 (100) NOT NULL
Phone VARCHAR2 (20) NOT NULL
CustomerType CHAR (1) NOT NULL
ACCOUNT
PK AccountNo NUMBER (15) PRIMARY KEY
PresentBalance DEC (10,2) NOT NULL
AvailableBalance DEC (10,2) NOT NULL
FK AccountStatusID NUMBER (2) NOT NULL
AccountType CHAR (1) NOT NULL
ACCOUNT_STATUS
PK AccountStatusID NUMBER (2) PRIMARY KEY
AccountStatusDesc VARCHAR2 (15) NOT NULL
AUTO
PK ALoanNo NUMBER (10) PRIMARY KEY
DealerName VARCHAR2 (80) NOT NULL
Number-Street VARCHAR2 (100) NOT NULL
City VARCHAR2 (75) NOT NULL
State VARCHAR2 (50) NOT NULL
Zip VARCHAR2 (10) NOT NULL
Country VARCHAR2 (75) NOT NULL
Made VARCHAR2 (20) NOT NULL
Model VARCHAR2 (20) NOT NULL
Year NUMBER (4) NOT NULL
PurchaseStatus VARCHAR (9) NOT NULL
VehiclePrice DEC (9,2) NOT NULL
DownPayment DEC (9,2) NOT NULL
FK VehicleTypeID NUMBER (2) NOT NULL
STUDENT
PK SLoanNo NUMBER (10) PRIMARY KEY
SchoolName VARCHAR2 (80) NOT NULL
Number-Street VARCHAR2 (100) NOT NULL
City VARCHAR2 (75) NOT NULL
State VARCHAR2 (50) NOT NULL
Zip VARCHAR2 (10) NOT NULL
Country VARCHAR2 (75) NOT NULL
AcademicPeriod VARCHAR2 (50) NOT NULL
Major VARCHAR2 (80) NOT NULL
GradDate DATE NOT NULL
SAVINGS
PK SAccountNo NUMBER (15) PRIMARY KEY
SavingsInterestRate DEC (3,2) NOT NULL
ExcessiveWithdrawCharge DEC (4,2) NOT NULL
CHECKING
PK CAccountNo NUMBER (15) PRIMARY KEY
ServiceCharge DEC (4,2) NOT NULL
OverdraftProtectionStatus VARCHAR2 (11) NOT NULL
ProtectionLimit DEC (10,2) NULL
PERSON
PK PCustomerID NUMBER (5) PRIMARY KEY
PersonFirst VARCHAR2 (50) NOT NULL
PersonLast VARCHAR2 (50) NOT NULL
PersonSSN VARCHAR2 (20) NOT NULL, UNIQUE
PersonDOB DATE NOT NULL
Mobile VARCHAR2 (20) NOT NULL
Gender CHAR (1) NOT NULL
STUDENT_DEGREE
PK SLoanNo NUMBER (10) PRIMARY KEY
PK DegreeName VARCHAR2 (80) PRIMARY KEY
PK SchoolFrom VARCHAR2 (80) PRIMARY KEY
PK DateObtained DATE PRIMARY KEY
CORPORATE
PK CCustomerID NUMBER (5) PRIMARY KEY
FK CompanyTaxID VARCHAR2 (20) NOT NULL
ContactName VARCHAR2 (75) NOT NULL
ContactPhone VARCHAR2 (20) NOT NULL
ContactEmail VARCHAR2 (100) NOT NULL
FK BankerID NUMBER (4) NOT NULL
BANKER
PK BankerID NUMBER (4) PRIMARY KEY
BankerFirst VARCHAR2 (50) NOT NULL
BankerLast VARCHAR2 (50) NOT NULL
BankerPhone VARCHAR2 (20) NOT NULL
BankerEmail VARCHAR2 (70) NOT NULL
VEHICLE_TYPE
PK VehicleTypeID NUMBER (2) PRIMARY KEY
VehicleTypeDesc VARCHAR2 (30) NOT NULL
PERSON_BENEFICIARY
PK PCustomerID NUMBER (5) PRIMARY KEY
PK BeneficiarySSN VARCHAR2 (20) PRIMARY KEY
COMPANY
PK CompanyTaxID VARCHAR2 (20) PRIMARY KEY
CompanyName VARCHAR2 (80) NOT NULL
BENEFICIARY
PK BeneficiarySSN VARCHAR2 (20) PRIMARY KEY
FirstName VARCHAR2 (50) NOT NULL
LastName VARCHAR2 (50) NOT NULL
DOB DATE NOT NULL

## Development & Implementation Phase – Implementation of Physical Model

In this phase, the entire physical design will be implemented into Oracle database by creating tables and relationships using a script file containing all the CREATE TABLE statements as shown below.

```
CREATE TABLE Location
(
   LocationID      NUMBER(4)       PRIMARY KEY,
   Number_Street   VARCHAR2(100)   NOT NULL,
   City           VARCHAR2(75)    NOT NULL,
   State          VARCHAR2(50)    NOT NULL,
   Zip            VARCHAR2(10)    NOT NULL,
   Country        VARCHAR2(75)    NOT NULL,
   BranchYN       CHAR(1)         NOT NULL,
   ATMYN          CHAR(1)         NOT NULL
);
CREATE TABLE Branch
(
   BLocationID      NUMBER(4)      PRIMARY KEY,
   BranchNo        NUMBER(4)           NOT NULL UNIQUE,
   BranchName           VARCHAR2(70)    NOT NULL,
   ManagerFirst          VARCHAR2(50)    NOT NULL,
   ManagerLast           VARCHAR2(50)    NOT NULL,
   BranchPhone        VARCHAR2(20)    NOT NULL,
   BranchEmail      VARCHAR2(70)        NOT NULL,
   CONSTRAINT BLocationID_FK FOREIGN KEY (BLocationID) REFERENCES Location(LocationID)
);
CREATE TABLE ATM
(
   ALocationID      NUMBER(4)      PRIMARY KEY,
   ATMNo           NUMBER(4)      NOT NULL,
   CashLimit       NUMBER(4)      NOT NULL,
   CONSTRAINT ALocationID_FK FOREIGN KEY (ALocationID) REFERENCES Location(LocationID),
   CONSTRAINT ATMNo_Unique UNIQUE (ATMNo)
);
CREATE TABLE Customer
(
   CustomerID      NUMBER(5)       PRIMARY KEY,
   Number_Street   VARCHAR2(100)   NOT NULL,
   City           VARCHAR2(75)    NOT NULL,
   State          VARCHAR2(50)    NOT NULL,
   Zip            VARCHAR2(10)    NOT NULL,
   Country        VARCHAR2(75)    NOT NULL,
   Email          VARCHAR2(100)   NOT NULL,
   Phone          VARCHAR2(20)    NOT NULL,
   CustomerType    CHAR(1)         NOT NULL
);
CREATE TABLE Person
(
   PCustomerID      NUMBER(5)       PRIMARY KEY,
   PersonFirst     VARCHAR2(50)    NOT NULL,
   PersonLast      VARCHAR2(50)    NOT NULL,
```

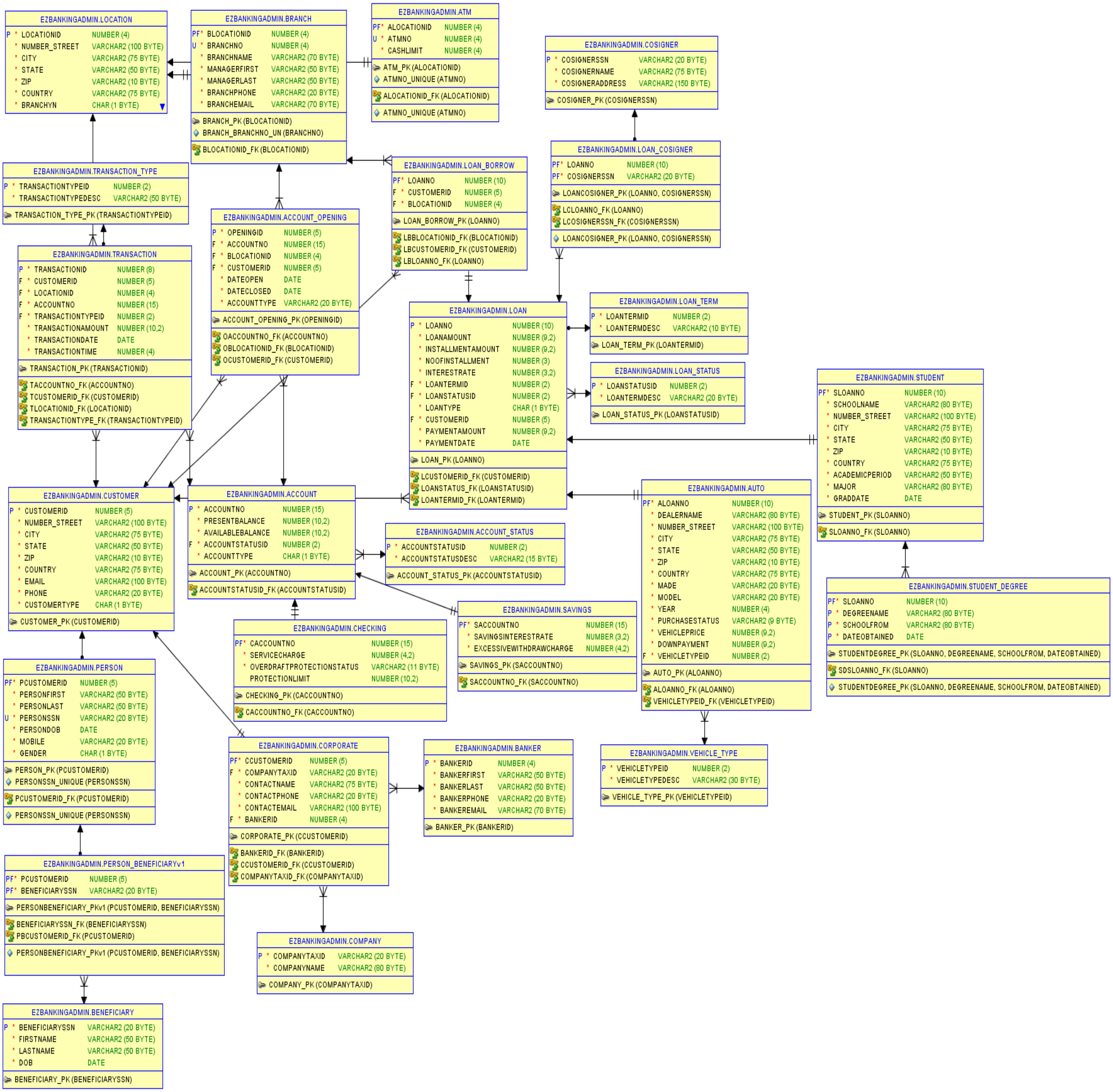
```sql
   PersonSSN       VARCHAR2(20)    NOT NULL,
   PersonDOB       DATE            NOT NULL,
   Mobile          VARCHAR2(20)    NOT NULL,
   Gender          CHAR(1)         NOT NULL,
   CONSTRAINT PCustomerID_FK FOREIGN KEY (PCustomerID) REFERENCES Customer(CustomerID),
   CONSTRAINT PersonSSN_Unique UNIQUE (PersonSSN)
);
CREATE TABLE Beneficiary
(
   BeneficiarySSN  VARCHAR2(20)    PRIMARY KEY,
   FirstName       VARCHAR2(50)    NOT NULL,
   LastName        VARCHAR2(50)    NOT NULL,
   DOB             DATE            NOT NULL
);
CREATE TABLE Person_Beneficiary
(
   PCustomerID     NUMBER(5),
   BeneficiarySSN  VARCHAR2(20),
   CONSTRAINT PersonBeneficiary_PK PRIMARY KEY (PCustomerID,BeneficiarySSN),
   CONSTRAINT PBCustomerID_FK FOREIGN KEY (PCustomerID) REFERENCES Person(PCustomerID),
   CONSTRAINT BeneficiarySSN_FK FOREIGN KEY (BeneficiarySSN) REFERENCES Beneficiary(BeneficiarySSN)
);
CREATE TABLE Company
(
   CompanyTaxID    VARCHAR2(20)    PRIMARY KEY,
   CompanyName     VARCHAR2(80)    NOT NULL
);
CREATE TABLE Banker
(
   BankerID        NUMBER(4)       PRIMARY KEY,
   BankerFirst     VARCHAR2(50)    NOT NULL,
   BankerLast      VARCHAR2(50)    NOT NULL,
   BankerPhone     VARCHAR2(20)    NOT NULL,
   BankerEmail     VARCHAR2(70)    NOT NULL
);
CREATE TABLE Corporate
(
   CCustomerID     NUMBER(5)       PRIMARY KEY,
   CompanyTaxID    VARCHAR2(20)    NOT NULL,
   ContactName     VARCHAR2(75)    NOT NULL,
   ContactPhone    VARCHAR2(20)    NOT NULL,
   ContactEmail    VARCHAR2(100)   NOT NULL,
   BankerID        NUMBER(4)       NOT NULL,
   CONSTRAINT CCustomerID_FK FOREIGN KEY (CCustomerID) REFERENCES Customer(CustomerID),
   CONSTRAINT CompanyTaxID_FK FOREIGN KEY (CompanyTaxID) REFERENCES Company(CompanyTaxID),
   CONSTRAINT BankerID_FK  FOREIGN KEY (BankerID) REFERENCES Banker (BankerID)
);
CREATE TABLE Account_Status
(
   AccountStatusID    NUMBER(2)     PRIMARY KEY,
   AccountStatusDesc  VARCHAR2(15)  NOT NULL
);
```

```
CREATE TABLE Account
(
   AccountNo       NUMBER(15)      PRIMARY KEY,
   PresentBalance  DEC(10,2)       NOT NULL,
   AvailableBalance    DEC(10,2)   NOT NULL,
   AccountStatusID     NUMBER(2)   NOT NULL,
   AccountType     CHAR(1)         NOT NULL,
   CONSTRAINT AccountStatusID_FK FOREIGN KEY (AccountStatusID) REFERENCES Account_Status
(AccountStatusID)
);
CREATE TABLE Checking
(
   CAccountNo       NUMBER(15)      PRIMARY KEY,
   ServiceCharge    DEC(4,2)        NOT NULL,
   OverdraftProtectionStatus    VARCHAR2(11)   NOT NULL,
   ProtectionLimit             DEC(10,2)      NULL,
   CONSTRAINT CAccountNo_FK FOREIGN KEY (CAccountNo) REFERENCES Account (AccountNo)
);
CREATE TABLE Savings
(
   SAccountNo            NUMBER(15)      PRIMARY KEY,
   SavingsInterestRate    DEC(3,2)        NOT NULL,
   ExcessiveWithdrawCharge DEC(4,2)       NOT NULL,
   CONSTRAINT SAccountNo_FK FOREIGN KEY (SAccountNo) REFERENCES Account (AccountNo)
);
CREATE TABLE Account_Opening
(
   OpeningID      NUMBER(5)       PRIMARY KEY,
   AccountNo      NUMBER(15)      NOT NULL,
   BLocationID     NUMBER(4)       NOT NULL,
   CustomerID     NUMBER(5)       NOT NULL,
   DateOpen       DATE            NOT NULL,
   DateClosed      DATE            NOT NULL,
   AccountType     VARCHAR2(20)    NOT NULL,
   CONSTRAINT OAccountNo_FK FOREIGN KEY (AccountNo) REFERENCES Account (AccountNo),
   CONSTRAINT OBLocationID_FK FOREIGN KEY (BLocationID) REFERENCES Branch (BLocationID),
   CONSTRAINT OCustomerID_FK FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID)
);
CREATE TABLE Transaction_Type
(
   TransactionTypeID   NUMBER(2)     PRIMARY KEY,
   TransactionTypeDesc VARCHAR2(50)  NOT NULL
);
CREATE TABLE Transaction
(
   TransactionID       NUMBER(8)       PRIMARY KEY,
   CustomerID          NUMBER(5)       NOT NULL,
   LocationID         NUMBER(4)       NOT NULL,
   AccountNo          NUMBER(15)      NOT NULL,
   TransactionTypeID   NUMBER(2)       NOT NULL,
   TransactionAmount   DEC(10,2)       NOT NULL,
   TransactionDate     DATE            NOT NULL,
  TransactionTime    NUMBER(4)      NOT NULL,
  CONSTRAINT TAccountNo_FK FOREIGN KEY (AccountNo) REFERENCES Account (AccountNo),
  CONSTRAINT TLocationID_FK FOREIGN KEY (LocationID) REFERENCES Location (LocationID),
  CONSTRAINT TCustomerID_FK FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID),
  CONSTRAINT TransactionType_FK FOREIGN KEY (TransactionTypeID) REFERENCES
Transaction_Type(TransactionTypeID)
);
CREATE TABLE Loan_Term
(
  LoanTermID        NUMBER(2)   PRIMARY KEY,
  LoanTermDesc       VARCHAR2(10)NOT NULL
);
CREATE TABLE Loan_Status
(
  LoanStatusID       NUMBER(2)   PRIMARY KEY,
  LoanTermDesc       VARCHAR2(20)NOT NULL
);
CREATE TABLE Loan
(
  LoanNo            NUMBER(10)  PRIMARY KEY,
  LoanAmount         DEC(9,2)    NOT NULL,
  InstallmentAmount   DEC(9,2)    NOT NULL,
  NoOfInstallment    NUMBER(3)   NOT NULL,
  InterestRate       DEC(3,2)    NOT NULL,
  LoanTermID         NUMBER(2)   NOT NULL,
  LoanStatusID       NUMBER(2)   NOT NULL,
  LoanType           CHAR(1)     NOT NULL,
  CustomerID         NUMBER(5)   NOT NULL,
  PaymentAmount      DEC(9,2)    NOT NULL,
  PaymentDate        DATE        NOT NULL,
  CONSTRAINT LCustomerID_FK FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID),
  CONSTRAINT LoanTermID_FK FOREIGN KEY (LoanTermID) REFERENCES Loan_Term (LoanTermID),
  CONSTRAINT LoanStatus_FK FOREIGN KEY (LoanStatusID) REFERENCES Loan_Status (LoanStatusID)
);
CREATE TABLE Cosigner
(
  CosignerSSN     VARCHAR2(20)    PRIMARY KEY,
  CosignerName    VARCHAR2(75)    NOT NULL,
  CosignerAddress VARCHAR2(150)   NOT NULL
);
CREATE TABLE Loan_Cosigner
(
  LoanNo        NUMBER(10),
  CosignerSSN     VARCHAR2(20),
  CONSTRAINT LoanCosigner_PK PRIMARY KEY (LoanNo,CosignerSSN),
  CONSTRAINT LCosignerSSN_FK FOREIGN KEY (CosignerSSN) REFERENCES Cosigner (CosignerSSN),
  CONSTRAINT LCLoanNo_FK FOREIGN KEY (LoanNo) REFERENCES Loan (LoanNo)
);
CREATE TABLE Vehicle_Type
(
  VehicleTypeID   NUMBER(2)      PRIMARY KEY,
  VehicleTypeDesc VARCHAR2(30)    NOT NULL
```

```
);
CREATE TABLE Auto
(
   ALoanNo        NUMBER(10)      PRIMARY KEY,
   DealerName      VARCHAR2(80)    NOT NULL,
   Number_Street   VARCHAR2(100)   NOT NULL,
   City           VARCHAR2(75)    NOT NULL,
   State          VARCHAR2(50)    NOT NULL,
   Zip            VARCHAR2(10)    NOT NULL,
   Country        VARCHAR2(75)    NOT NULL,
   Made           VARCHAR2(20)    NOT NULL,
   Model          VARCHAR2(20)    NOT NULL,
   Year           NUMBER(4)       NOT NULL,
   PurchaseStatus  VARCHAR2(9)     NOT NULL,
   VehiclePrice    DEC(9,2)        NOT NULL,
   DownPayment     DEC(9,2)        NOT NULL,
   VehicleTypeID   NUMBER(2)       NOT NULL,
   CONSTRAINT ALoanNo_FK FOREIGN KEY (ALoanNo) REFERENCES Loan (LoanNo),
   CONSTRAINT VehicleTypeID_FK FOREIGN KEY (VehicleTypeID) REFERENCES Vehicle_Type (VehicleTypeID)
);
CREATE TABLE Student
(
   SLoanNo        NUMBER(10)      PRIMARY KEY,
   SchoolName      VARCHAR2(80)    NOT NULL,
   Number_Street   VARCHAR2(100)   NOT NULL,
   City           VARCHAR2(75)    NOT NULL,
   State          VARCHAR2(50)    NOT NULL,
   Zip            VARCHAR2(10)    NOT NULL,
   Country        VARCHAR2(75)    NOT NULL,
   AcademicPeriod  VARCHAR2(50)    NOT NULL,
   Major          VARCHAR2(80)    NOT NULL,
   GradDate       DATE            NOT NULL,
   CONSTRAINT SLoanNo_FK FOREIGN KEY (SLoanNo) REFERENCES Loan (LoanNo)
);
CREATE TABLE Student_Degree
(
   SLoanNo        NUMBER(10),
   DegreeName      VARCHAR2(80),
   SchoolFrom      VARCHAR2(80),
   DateObtained    DATE,
   CONSTRAINT StudentDegree_PK PRIMARY KEY (SLoanNo,DegreeName,SchoolFrom,DateObtained),
   CONSTRAINT SDSLoanNo_FK FOREIGN KEY (SLoanNo) REFERENCES Student (SLoanNo)
);
CREATE TABLE Loan_Borrow
(
   LoanNo         NUMBER(10)  PRIMARY KEY,
   CustomerID      NUMBER(5)   NOT NULL,
   BLocationID     NUMBER(4)   NOT NULL,
   CONSTRAINT LBBLocationID_FK FOREIGN KEY (BLocationID) REFERENCES Branch (BLocationID),
   CONSTRAINT LBCustomerID_FK FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID),
   CONSTRAINT LBLoanNo_FK FOREIGN KEY (LoanNo) REFERENCES Loan(LoanNo));
```

# Implementation Phase – Table Schema Diagram

This phase will generate a table diagram of the Physical Model that was created before in SQL Developer. The deliverable is shown below.



EZBANKINGADMIN.LOCATION

| | | |
|---|---|---|
| P * | LOCATIONID | NUMBER (4) |
| * | NUMBER_STREET | VARCHAR2 (100 BYTE) |
| * | CITY | VARCHAR2 (75 BYTE) |
| * | STATE | VARCHAR2 (50 BYTE) |
| * | ZIP | VARCHAR2 (10 BYTE) |
| * | COUNTRY | VARCHAR2 (75 BYTE) |
| * | BRANCHYN | CHAR (1 BYTE) |

EZBANKINGADMIN.BRANCH

| | | |
|---|---|---|
| PF* | BLOCATIONID | NUMBER (4) |
| U * | BRANCHNO | NUMBER (4) |
| * | BRANCHNAME | VARCHAR2 (70 BYTE) |
| * | MANAGERFIRST | VARCHAR2 (50 BYTE) |
| * | MANAGERLAST | VARCHAR2 (50 BYTE) |
| * | BRANCHPHONE | VARCHAR2 (20 BYTE) |
| * | BRANCHEMAIL | VARCHAR2 (70 BYTE) |

BRANCH_PK (BLOCATIONID)
BRANCH_BRANCHNO_UN (BRANCHNO)

BLOCATIONID_FK (BLOCATIONID)

EZBANKINGADMIN.ATM

| | | |
|---|---|---|
| PF* | ALOCATIONID | NUMBER (4) |
| U * | ATMNO | NUMBER (4) |
| * | CASHLIMIT | NUMBER (4) |

ATM_PK (ALOCATIONID)
ATMNO_UNIQUE (ATMNO)

ALOCATIONID_FK (ALOCATIONID)

ATMNO_UNIQUE (ATMNO)

EZBANKINGADMIN.COSIGNER

| | | |
|---|---|---|
| P * | COSIGNERSSN | VARCHAR2 (20 BYTE) |
| * | COSIGNERNAME | VARCHAR2 (75 BYTE) |
| * | COSIGNERADDRESS | VARCHAR2 (150 BYTE) |

COSIGNER_PK (COSIGNERSSN)

EZBANKINGADMIN.TRANSACTION_TYPE

| | | |
|---|---|---|
| P * | TRANSACTIONTYPEID | NUMBER (2) |
| * | TRANSACTIONTYPEDESC | VARCHAR2 (50 BYTE) |

TRANSACTION_TYPE_PK (TRANSACTIONTYPEID)

EZBANKINGADMIN.LOAN_BORROW

| | | |
|---|---|---|
| PF* | LOANNO | NUMBER (10) |
| F * | CUSTOMERID | NUMBER (5) |
| F * | BLOCATIONID | NUMBER (4) |

LOAN_BORROW_PK (LOANNO)

LBBLOCATIONID_FK (BLOCATIONID)
LBCUSTOMERID_FK (CUSTOMERID)
LBLOANNO_FK (LOANNO)

EZBANKINGADMIN.LOAN_COSIGNER

| | | |
|---|---|---|
| PF* | LOANNO | NUMBER (10) |
| PF* | COSIGNERSSN | VARCHAR2 (20 BYTE) |

LOANCOSIGNER_PK (LOANNO, COSIGNERSSN)

LCLOANNO_FK (LOANNO)
LCOSIGNERSSN_FK (COSIGNERSSN)

LOANCOSIGNER_PK (LOANNO, COSIGNERSSN)

EZBANKINGADMIN.ACCOUNT_OPENING

| | | |
|---|---|---|
| P * | OPENINGID | NUMBER (5) |
| F * | ACCOUNTNO | NUMBER (15) |
| F * | BLOCATIONID | NUMBER (4) |
| F * | CUSTOMERID | NUMBER (5) |
| * | DATEOPEN | DATE |
| * | DATECLOSED | DATE |
| * | ACCOUNTTYPE | VARCHAR2 (20 BYTE) |

ACCOUNT_OPENING_PK (OPENINGID)

OACCOUNTNO_FK (ACCOUNTNO)
OBLOCATIONID_FK (BLOCATIONID)
OCUSTOMERID_FK (CUSTOMERID)

EZBANKINGADMIN.TRANSACTION

| | | |
|---|---|---|
| P * | TRANSACTIONID | NUMBER (8) |
| F * | CUSTOMERID | NUMBER (5) |
| F * | LOCATIONID | NUMBER (4) |
| F * | ACCOUNTNO | NUMBER (15) |
| F * | TRANSACTIONTYPEID | NUMBER (2) |
| * | TRANSACTIONAMOUNT | NUMBER (10,2) |
| * | TRANSACTIONDATE | DATE |
| * | TRANSACTIONTIME | NUMBER (4) |

TRANSACTION_PK (TRANSACTIONID)

TACCOUNTNO_FK (ACCOUNTNO)
TCUSTOMERID_FK (CUSTOMERID)
TLOCATIONID_FK (LOCATIONID)
TRANSACTIONTYPE_FK (TRANSACTIONTYPEID)

EZBANKINGADMIN.LOAN

| | | |
|---|---|---|
| P * | LOANNO | NUMBER (10) |
| * | LOANAMOUNT | NUMBER (9,2) |
| * | INSTALLMENTAMOUNT | NUMBER (9,2) |
| * | NOOFINSTALLMENT | NUMBER (3) |
| * | INTERESTRATE | NUMBER (3,2) |
| F * | LOANTERMID | NUMBER (2) |
| F * | LOANSTATUSID | NUMBER (2) |
| * | LOANTYPE | CHAR (1 BYTE) |
| F * | CUSTOMERID | NUMBER (5) |
| * | PAYMENTAMOUNT | NUMBER (9,2) |
| * | PAYMENTDATE | DATE |

LOAN_PK (LOANNO)

LCUSTOMERID_FK (CUSTOMERID)
LOANSTATUS_FK (LOANSTATUSID)
LOANTERMID_FK (LOANTERMID)

EZBANKINGADMIN.LOAN_TERM

| | | |
|---|---|---|
| P * | LOANTERMID | NUMBER (2) |
| * | LOANTERMDESC | VARCHAR2 (10 BYTE) |

LOAN_TERM_PK (LOANTERMID)

EZBANKINGADMIN.LOAN_STATUS

| | | |
|---|---|---|
| P * | LOANSTATUSID | NUMBER (2) |
| * | LOANTERMDESC | VARCHAR2 (20 BYTE) |

LOAN_STATUS_PK (LOANSTATUSID)

EZBANKINGADMIN.STUDENT

| | | |
|---|---|---|
| PF* | SLOANNO | NUMBER (10) |
| * | SCHOOLNAME | VARCHAR2 (80 BYTE) |
| * | NUMBER_STREET | VARCHAR2 (100 BYTE) |
| * | CITY | VARCHAR2 (75 BYTE) |
| * | STATE | VARCHAR2 (50 BYTE) |
| * | ZIP | VARCHAR2 (10 BYTE) |
| * | COUNTRY | VARCHAR2 (75 BYTE) |
| * | ACADEMICPERIOD | VARCHAR2 (50 BYTE) |
| * | MAJOR | VARCHAR2 (80 BYTE) |
| * | GRADDATE | DATE |

STUDENT_PK (SLOANNO)

SLOANNO_FK (SLOANNO)

EZBANKINGADMIN.CUSTOMER

| | | |
|---|---|---|
| P * | CUSTOMERID | NUMBER (5) |
| * | NUMBER_STREET | VARCHAR2 (100 BYTE) |
| * | CITY | VARCHAR2 (75 BYTE) |
| * | STATE | VARCHAR2 (50 BYTE) |
| * | ZIP | VARCHAR2 (10 BYTE) |
| * | COUNTRY | VARCHAR2 (75 BYTE) |
| * | EMAIL | VARCHAR2 (100 BYTE) |
| * | PHONE | VARCHAR2 (20 BYTE) |
| * | CUSTOMERTYPE | CHAR (1 BYTE) |

CUSTOMER_PK (CUSTOMERID)

EZBANKINGADMIN.ACCOUNT

| | | |
|---|---|---|
| P * | ACCOUNTNO | NUMBER (15) |
| * | PRESENTBALANCE | NUMBER (10,2) |
| * | AVAILABLEBALANCE | NUMBER (10,2) |
| F * | ACCOUNTSTATUSID | NUMBER (2) |
| * | ACCOUNTTYPE | CHAR (1 BYTE) |

ACCOUNT_PK (ACCOUNTNO)

ACCOUNTSTATUSID_FK (ACCOUNTSTATUSID)

EZBANKINGADMIN.ACCOUNT_STATUS

| | | |
|---|---|---|
| P * | ACCOUNTSTATUSID | NUMBER (2) |
| * | ACCOUNTSTATUSDESC | VARCHAR2 (15 BYTE) |

ACCOUNT_STATUS_PK (ACCOUNTSTATUSID)

EZBANKINGADMIN.AUTO

| | | |
|---|---|---|
| PF* | ALOANNO | NUMBER (10) |
| * | DEALERNAME | VARCHAR2 (80 BYTE) |
| * | NUMBER_STREET | VARCHAR2 (100 BYTE) |
| * | CITY | VARCHAR2 (75 BYTE) |
| * | STATE | VARCHAR2 (50 BYTE) |
| * | ZIP | VARCHAR2 (10 BYTE) |
| * | COUNTRY | VARCHAR2 (75 BYTE) |
| * | MADE | VARCHAR2 (20 BYTE) |
| * | MODEL | VARCHAR2 (20 BYTE) |
| * | YEAR | NUMBER (4) |
| * | PURCHASESTATUS | VARCHAR2 (9 BYTE) |
| * | VEHICLEPRICE | NUMBER (9,2) |
| * | DOWNPAYMENT | NUMBER (9,2) |
| F * | VEHICLETYPEID | NUMBER (2) |

AUTO_PK (ALOANNO)

ALOANNO_FK (ALOANNO)
VEHICLETYPEID_FK (VEHICLETYPEID)

EZBANKINGADMIN.STUDENT_DEGREE

| | | |
|---|---|---|
| PF* | SLOANNO | NUMBER (10) |
| P * | DEGREENAME | VARCHAR2 (80 BYTE) |
| P * | SCHOOLFROM | VARCHAR2 (80 BYTE) |
| P * | DATEOBTAINED | DATE |

STUDENTDEGREE_PK (SLOANNO, DEGREENAME, SCHOOLFROM, DATEOBTAINED)

SDSLOANNO_FK (SLOANNO)

STUDENTDEGREE_PK (SLOANNO, DEGREENAME, SCHOOLFROM, DATEOBTAINED)

EZBANKINGADMIN.SAVINGS

| | | |
|---|---|---|
| PF* | SACCOUNTNO | NUMBER (15) |
| * | SAVINGSINTERESTRATE | NUMBER (3,2) |
| * | EXCESSIVEWITHDRAWCHARGE | NUMBER (4,2) |

SAVINGS_PK (SACCOUNTNO)

SACCOUNTNO_FK (SACCOUNTNO)

EZBANKINGADMIN.CHECKING

| | | |
|---|---|---|
| PF* | CACCOUNTNO | NUMBER (15) |
| * | SERVICECHARGE | NUMBER (4,2) |
| * | OVERDRAFTPROTECTIONSTATUS | VARCHAR2 (11 BYTE) |
| | PROTECTIONLIMIT | NUMBER (10,2) |

CHECKING_PK (CACCOUNTNO)

CACCOUNTNO_FK (CACCOUNTNO)

EZBANKINGADMIN.PERSON

| | | |
|---|---|---|
| PF* | PCUSTOMERID | NUMBER (5) |
| * | PERSONFIRST | VARCHAR2 (50 BYTE) |
| * | PERSONLAST | VARCHAR2 (50 BYTE) |
| U * | PERSONSSN | VARCHAR2 (20 BYTE) |
| * | PERSONDOB | DATE |
| * | MOBILE | VARCHAR2 (20 BYTE) |
| * | GENDER | CHAR (1 BYTE) |

PERSON_PK (PCUSTOMERID)
PERSONSSN_UNIQUE (PERSONSSN)

PCUSTOMERID_FK (PCUSTOMERID)

PERSONSSN_UNIQUE (PERSONSSN)

EZBANKINGADMIN.CORPORATE

| | | |
|---|---|---|
| PF* | CUSTOMERID | NUMBER (5) |
| F * | COMPANYTAXID | VARCHAR2 (20 BYTE) |
| * | CONTACTNAME | VARCHAR2 (75 BYTE) |
| * | CONTACTPHONE | VARCHAR2 (20 BYTE) |
| * | CONTACTEMAIL | VARCHAR2 (100 BYTE) |
| F * | BANKERID | NUMBER (4) |

CORPORATE_PK (CCUSTOMERID)

BANKERID_FK (BANKERID)
CCUSTOMERID_FK (CCUSTOMERID)
COMPANYTAXID_FK (COMPANYTAXID)

EZBANKINGADMIN.BANKER

| | | |
|---|---|---|
| P * | BANKERID | NUMBER (4) |
| * | BANKERFIRST | VARCHAR2 (50 BYTE) |
| * | BANKERLAST | VARCHAR2 (50 BYTE) |
| * | BANKERPHONE | VARCHAR2 (20 BYTE) |
| * | BANKEREMAIL | VARCHAR2 (70 BYTE) |

BANKER_PK (BANKERID)

EZBANKINGADMIN.VEHICLE_TYPE

| | | |
|---|---|---|
| P * | VEHICLETYPEID | NUMBER (2) |
| * | VEHICLETYPEDESC | VARCHAR2 (30 BYTE) |

VEHICLE_TYPE_PK (VEHICLETYPEID)

EZBANKINGADMIN.PERSON_BENEFICIARYv1

| | | |
|---|---|---|
| PF* | PCUSTOMERID | NUMBER (5) |
| PF* | BENEFICIARYSSN | VARCHAR2 (20 BYTE) |

PERSONBENEFICIARY_PKv1 (PCUSTOMERID, BENEFICIARYSSN)

BENEFICIARYSSN_FK (BENEFICIARYSSN)
PBCUSTOMERID_FK (PCUSTOMERID)

PERSONBENEFICIARY_PKv1 (PCUSTOMERID, BENEFICIARYSSN)

EZBANKINGADMIN.COMPANY

| | | |
|---|---|---|
| P * | COMPANYTAXID | VARCHAR2 (20 BYTE) |
| * | COMPANYNAME | VARCHAR2 (80 BYTE) |

COMPANY_PK (COMPANYTAXID)

EZBANKINGADMIN.BENEFICIARY

| | | |
|---|---|---|
| P * | BENEFICIARYSSN | VARCHAR2 (20 BYTE) |
| * | FIRSTNAME | VARCHAR2 (50 BYTE) |
| * | LASTNAME | VARCHAR2 (50 BYTE) |
| * | DOB | DATE |

BENEFICIARY_PK (BENEFICIARYSSN)

## Implementation Phase – Testing and Validation

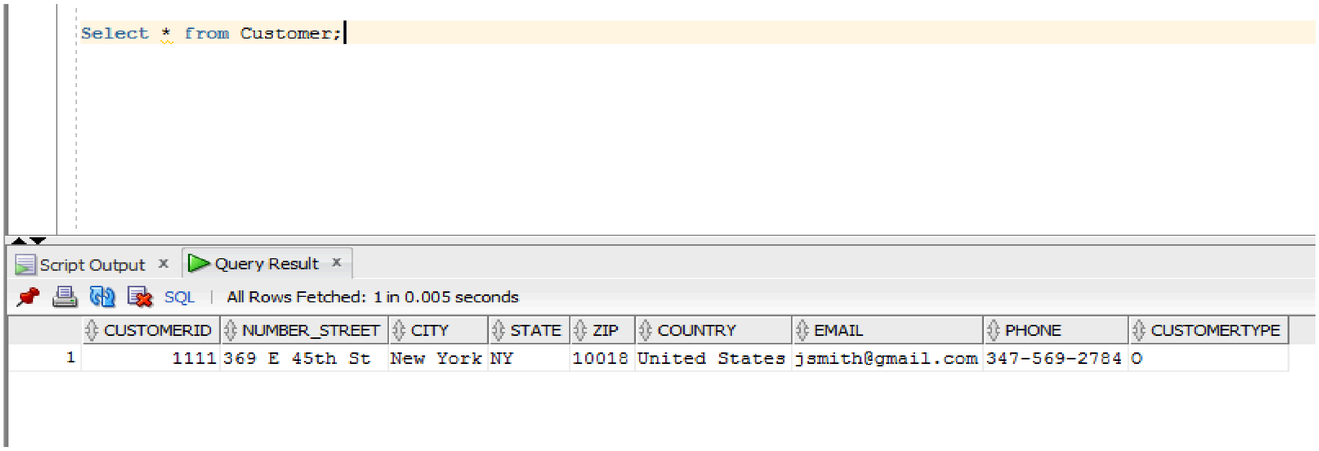
In this phase, a SQL script file will be created to test the banking transaction scenario including Insert, Select, Update, and Delete queries.

Tables participated in transaction scenario are Customer, Account, Location and Transaction (Note that Account_Status and Transaction_Type tables are required before inserting records into Account and Transaction tables).

INSERT STATEMENT

1. The following queries insert five records into Customer table.

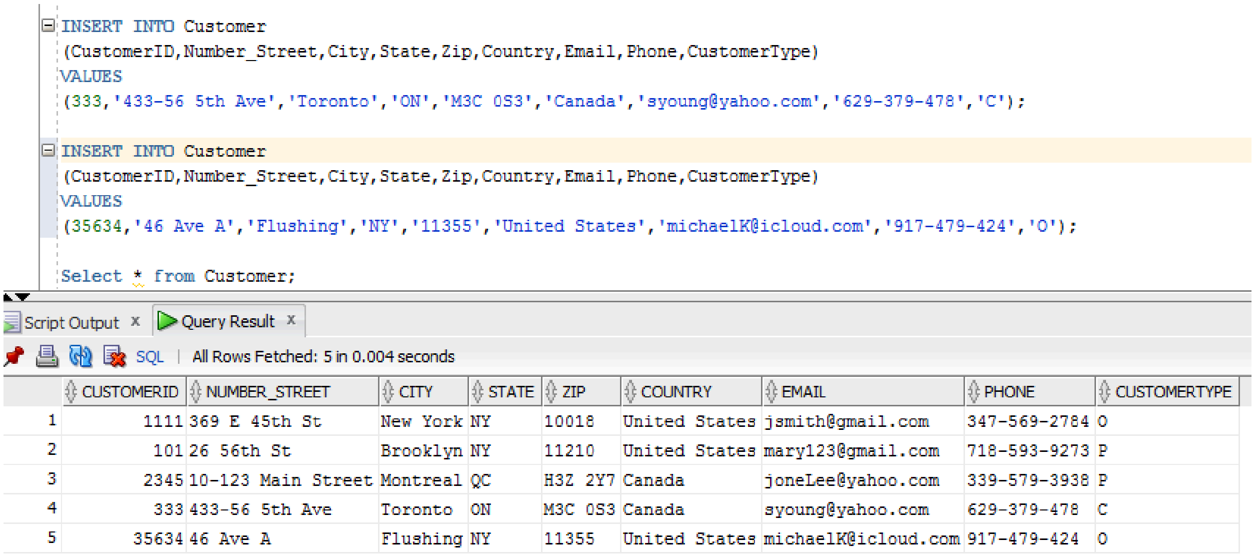
```
INSERT INTO Customer
(CustomerID,Number_Street,City,State,Zip,Country,Email,Phone,CustomerType)
VALUES
(1111,'369 E 45th St','New York','NY','10018','United States','jsmith@gmail.com','347-569-2784','O');

INSERT INTO Customer
(CustomerID,Number_Street,City,State,Zip,Country,Email,Phone,CustomerType)
VALUES
(101,'26 56th St','Brooklyn','NY','11210','United States','mary123@gmail.com','718-593-9273','P');

INSERT INTO Customer
(CustomerID,Number_Street,City,State,Zip,Country,Email,Phone,CustomerType)
VALUES
(2345,'10-123 Main Street','Montreal','QC','H3Z 2Y7','Canada','joneLee@yahoo.com','339-579-3938','P');

INSERT INTO Customer
(CustomerID,Number_Street,City,State,Zip,Country,Email,Phone,CustomerType)
VALUES
(333,'433-56 5th Ave','Toronto','ON','M3C 0S3','Canada','syoung@yahoo.com','629-379-478','C');

INSERT INTO Customer
(CustomerID,Number_Street,City,State,Zip,Country,Email,Phone,CustomerType)
VALUES
(35634,'46 Ave A','Flushing','NY','11355','United States','michaelK@icloud.com','917-479-424','O');
```



```
Select * from Customer;
```

All Rows Fetched: 1 in 0.005 seconds

| | CUSTOMERID | NUMBER_STREET | CITY | STATE | ZIP | COUNTRY | EMAIL | PHONE | CUSTOMERTYPE |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1111 | 369 E 45th St | New York | NY | 10018 | United States | jsmith@gmail.com | 347-569-2784 | O |



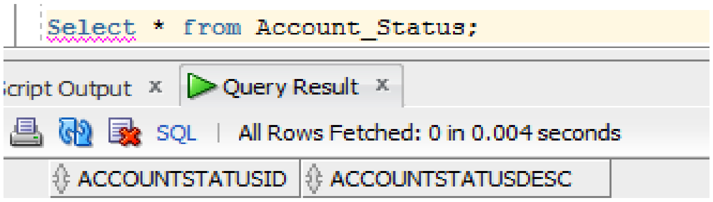
```
INSERT INTO Customer
(CustomerID,Number_Street,City,State,Zip,Country,Email,Phone,CustomerType)
VALUES
(333,'433-56 5th Ave','Toronto','ON','M3C 0S3','Canada','syoung@yahoo.com','629-379-478','C');

INSERT INTO Customer
(CustomerID,Number_Street,City,State,Zip,Country,Email,Phone,CustomerType)
VALUES
(35634,'46 Ave A','Flushing','NY','11355','United States','michaelK@icloud.com','917-479-424','O');

Select * from Customer;
```

All Rows Fetched: 5 in 0.004 seconds

| | CUSTOMERID | NUMBER_STREET | CITY | STATE | ZIP | COUNTRY | EMAIL | PHONE | CUSTOMERTYPE |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1111 | 369 E 45th St | New York | NY | 10018 | United States | jsmith@gmail.com | 347-569-2784 | O |
| 2 | 101 | 26 56th St | Brooklyn | NY | 11210 | United States | mary123@gmail.com | 718-593-9273 | P |
| 3 | 2345 | 10-123 Main Street | Montreal | QC | H3Z 2Y7 | Canada | joneLee@yahoo.com | 339-579-3938 | P |
| 4 | 333 | 433-56 5th Ave | Toronto | ON | M3C 0S3 | Canada | syoung@yahoo.com | 629-379-478 | C |
| 5 | 35634 | 46 Ave A | Flushing | NY | 11355 | United States | michaelK@icloud.com | 917-479-424 | O |

2. The following queries insert five records into Account_Status table.

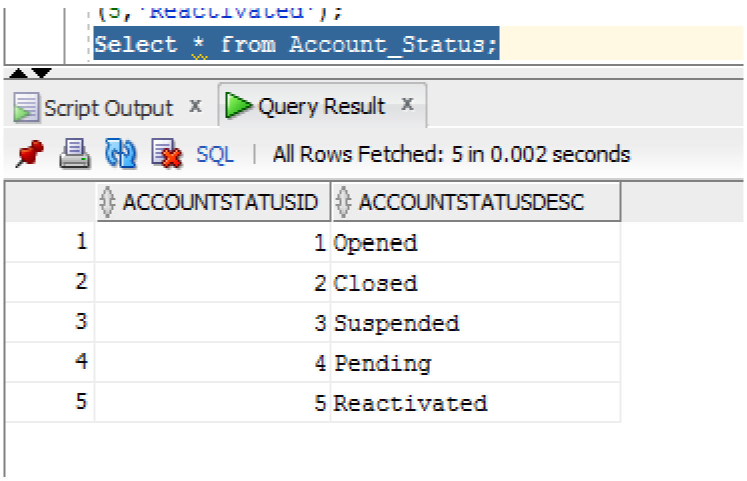
```
INSERT INTO Account_Status
(ACCOUNTSTATUSID,ACCOUNTSTATUSDESC)
VALUES
(1,'Opened');

INSERT INTO Account_Status
(ACCOUNTSTATUSID,ACCOUNTSTATUSDESC)
VALUES
(2,'Closed');

INSERT INTO Account_Status
(ACCOUNTSTATUSID,ACCOUNTSTATUSDESC)
VALUES
(3,'Suspended');

INSERT INTO Account_Status
(ACCOUNTSTATUSID,ACCOUNTSTATUSDESC)
VALUES
(4,'Pending');

INSERT INTO Account_Status
(ACCOUNTSTATUSID,ACCOUNTSTATUSDESC)
VALUES
(5,'Reactivated');
```



```
Select * from Account_Status;
```

All Rows Fetched: 0 in 0.004 seconds

| ACCOUNTSTATUSID | ACCOUNTSTATUSDESC |
|---|---|



```
(5,'Reactivated');
Select * from Account_Status;
```

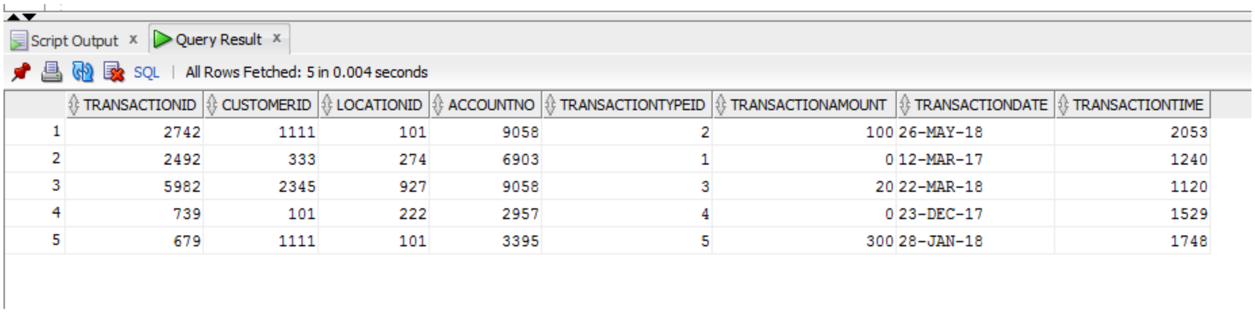
All Rows Fetched: 5 in 0.002 seconds

| | ACCOUNTSTATUSID | ACCOUNTSTATUSDESC |
|---|---|---|
| 1 | 1 | Opened |
| 2 | 2 | Closed |
| 3 | 3 | Suspended |
| 4 | 4 | Pending |
| 5 | 5 | Reactivated |

3. The following queries insert five records into Transaction table.

```
INSERT INTO Transaction
(TransactionID,CustomerID,LocationID,AccountNo,TransactionTypeID,TransactionAmount,TransactionDate,TransactionTime)
VALUES
(679,1111,101,3395,5,300.00,to_date('01282018','MMDDYYYY'),1748);

INSERT INTO Transaction
(TransactionID,CustomerID,LocationID,AccountNo,TransactionTypeID,TransactionAmount,TransactionDate,TransactionTime)
VALUES
(739,101,222,2957,4,0.00,to_date('12232017','MMDDYYYY'),1529);

INSERT INTO Transaction
(TransactionID,CustomerID,LocationID,AccountNo,TransactionTypeID,TransactionAmount,TransactionDate,TransactionTime)
VALUES
(5982,2345,927,9058,3,20.00,to_date('03222018','MMDDYYYY'),1120);

INSERT INTO Transaction
(TransactionID,CustomerID,LocationID,AccountNo,TransactionTypeID,TransactionAmount,TransactionDate,TransactionTime)
VALUES
(2742,1111,101,9058,2,100.00,to_date('05262018','MMDDYYYY'),2053);


INSERT INTO Transaction
```

(TransactionID,CustomerID,LocationID,AccountNo,TransactionTypeID,TransactionAmount,TransactionDate,TransactionTime)
VALUES
(2492,333,274,6903,1,0.00,to_date('03122017','MMDDYYYY'),1240);



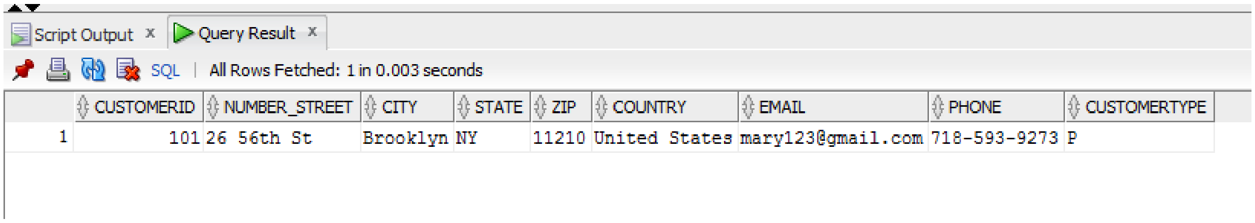
All Rows Fetched: 5 in 0.004 seconds

| | TRANSACTIONID | CUSTOMERID | LOCATIONID | ACCOUNTNO | TRANSACTIONTYPEID | TRANSACTIONAMOUNT | TRANSACTIONDATE | TRANSACTIONTIME |
|---|---|---|---|---|---|---|---|---|
| 1 | 2742 | 1111 | 101 | 9058 | 2 | 100 | 26-MAY-18 | 2053 |
| 2 | 2492 | 333 | 274 | 6903 | 1 | 0 | 12-MAR-17 | 1240 |
| 3 | 5982 | 2345 | 927 | 9058 | 3 | 20 | 22-MAR-18 | 1120 |
| 4 | 739 | 101 | 222 | 2957 | 4 | 0 | 23-DEC-17 | 1529 |
| 5 | 679 | 1111 | 101 | 3395 | 5 | 300 | 28-JAN-18 | 1748 |

SELECT STATEMENT

1. The following query returns information of customer with id = 101.

SELECT *
FROM Customer
Where CustomerID = 101;

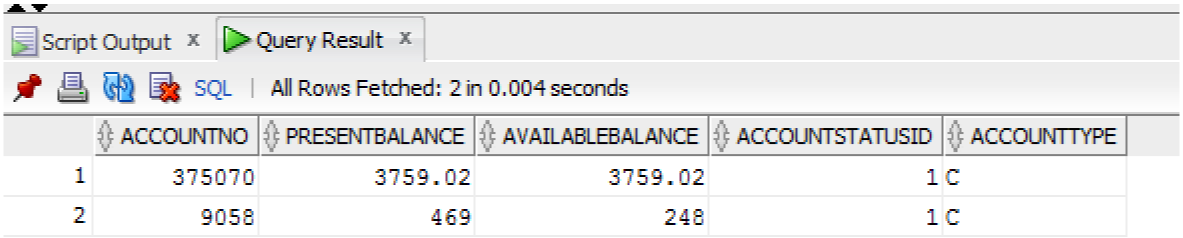


All Rows Fetched: 1 in 0.003 seconds

| | CUSTOMERID | NUMBER_STREET | CITY | STATE | ZIP | COUNTRY | EMAIL | PHONE | CUSTOMERTYPE |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 101 | 26 56th St | Brooklyn | NY | 11210 | United States | mary123@gmail.com | 718-593-9273 | P |

2. The following query returns information of all checking accounts.

SELECT *
FROM Account
WHERE AccountType = 'C';

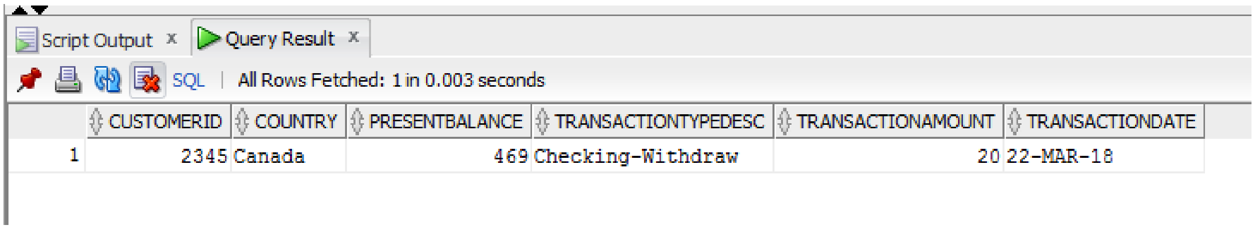


All Rows Fetched: 2 in 0.004 seconds

| | ACCOUNTNO | PRESENTBALANCE | AVAILABLEBALANCE | ACCOUNTSTATUSID | ACCOUNTTYPE |
|---|---|---|---|---|---|
| 1 | 375070 | 3759.02 | 3759.02 | 1 | C |
| 2 | 9058 | 469 | 248 | 1 | C |

3. The following query returns the id, country, account present balance and transaction information of a customer.

```
SELECT Transaction.CustomerID, Customer.Country, ACCOUNT.PRESENTBALANCE,
        TRANSACTION_TYPE.TRANSACTIONTYPEDESC,
        TRANSACTION.TRANSACTIONAMOUNT,TRANSACTION.TRANSACTIONDATE
FROM Customer, Account, Transaction_Type, Transaction
WHERE TRANSACTION.CUSTOMERID = 2345 AND Transaction.CUSTOMERID = Customer.CUSTOMERID
        AND Transaction.AccountNo = Account.ACCOUNTNO AND Transaction.TRANSACTIONTYPEID =
Transaction_Type.TRANSACTIONTYPEID;
```



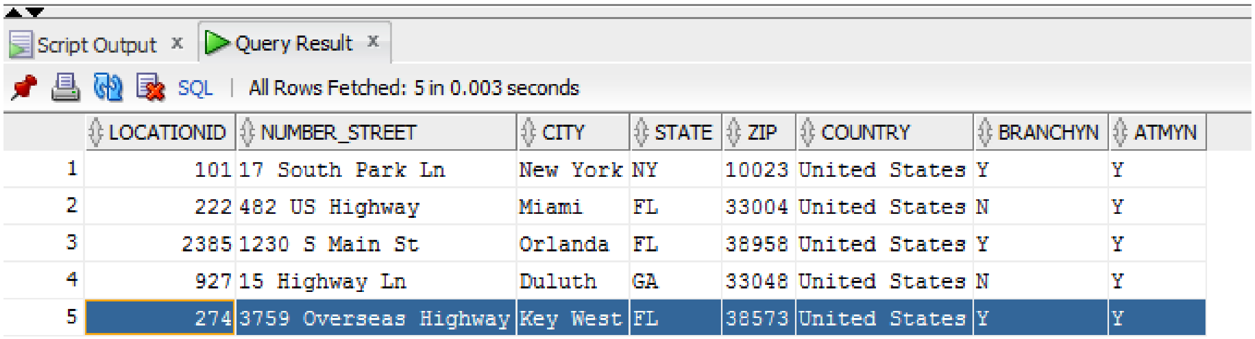
All Rows Fetched: 1 in 0.003 seconds

| | CUSTOMERID | COUNTRY | PRESENTBALANCE | TRANSACTIONTYPEDESC | TRANSACTIONAMOUNT | TRANSACTIONDATE |
|---|---|---|---|---|---|---|
| 1 | 2345 | Canada | 469 | Checking-Withdraw | 20 | 22-MAR-18 |

UPDATE STATEMENT

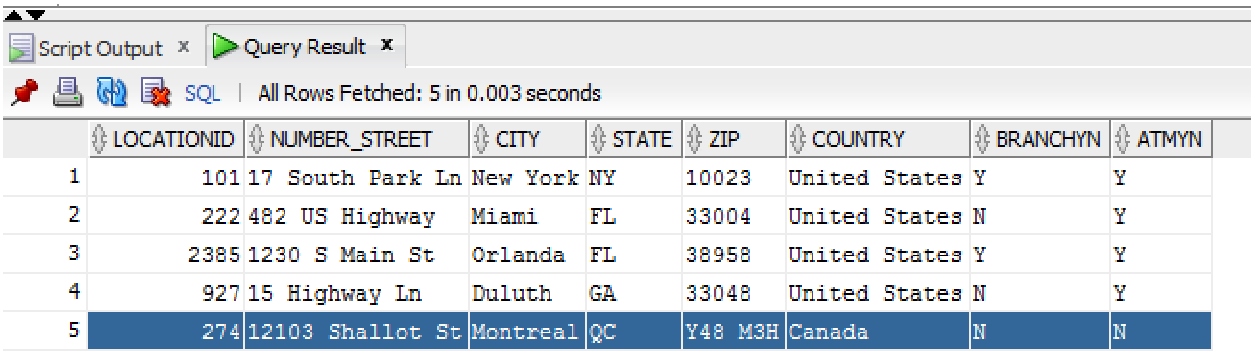
1. The following query updates the location (with location id = 274) with a new address and a new location type.

```
UPDATE Location
SET NUMBER_STREET = '12103 Shallot St',
      CITY = 'Montreal',
      STATE = 'QC',
      ZIP = 'Y48 M3H',
      COUNTRY = 'Canada',
      BRANCHYN = 'N',
     ATMYN = 'N'
WHERE LOCATIONID = 274;
```



All Rows Fetched: 5 in 0.003 seconds

| | LOCATIONID | NUMBER_STREET | CITY | STATE | ZIP | COUNTRY | BRANCHYN | ATMYN |
|---|---|---|---|---|---|---|---|---|
| 1 | 101 | 17 South Park Ln | New York | NY | 10023 | United States | Y | Y |
| 2 | 222 | 482 US Highway | Miami | FL | 33004 | United States | N | Y |
| 3 | 2385 | 1230 S Main St | Orlanda | FL | 38958 | United States | Y | Y |
| 4 | 927 | 15 Highway Ln | Duluth | GA | 33048 | United States | N | Y |
| 5 | 274 | 3759 Overseas Highway | Key West | FL | 38573 | United States | Y | Y |

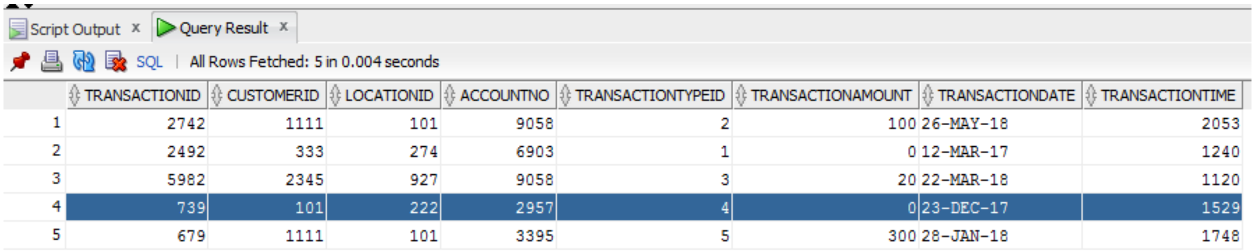


All Rows Fetched: 5 in 0.003 seconds

| | LOCATIONID | NUMBER_STREET | CITY | STATE | ZIP | COUNTRY | BRANCHYN | ATMYN |
|---|---|---|---|---|---|---|---|---|
| 1 | 101 | 17 South Park Ln | New York | NY | 10023 | United States | Y | Y |
| 2 | 222 | 482 US Highway | Miami | FL | 33004 | United States | N | Y |
| 3 | 2385 | 1230 S Main St | Orlanda | FL | 38958 | United States | Y | Y |
| 4 | 927 | 15 Highway Ln | Duluth | GA | 33048 | United States | N | Y |
| 5 | 274 | 12103 Shallot St | Montreal | QC | Y48 M3H | Canada | N | N |

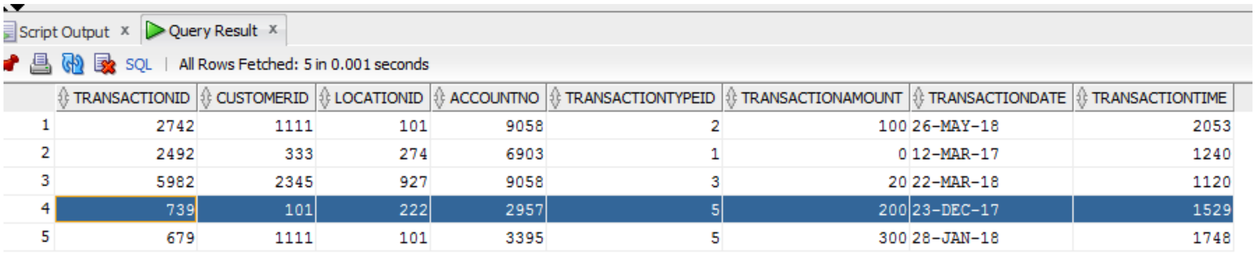
2. The following query updates the transaction record (with transaction id = 739) with a new transaction type and a new transaction amount.

```
UPDATE Transaction
SET TRANSACTIONTYPEID = 5,
     TRANSACTIONAMOUNT = 200.00
WHERE TRANSACTIONID = 739;
```



All Rows Fetched: 5 in 0.004 seconds

| | TRANSACTIONID | CUSTOMERID | LOCATIONID | ACCOUNTNO | TRANSACTIONTYPEID | TRANSACTIONAMOUNT | TRANSACTIONDATE | TRANSACTIONTIME |
|---|---|---|---|---|---|---|---|---|
| 1 | 2742 | 1111 | 101 | 9058 | 2 | 100 | 26-MAY-18 | 2053 |
| 2 | 2492 | 333 | 274 | 6903 | 1 | 0 | 12-MAR-17 | 1240 |
| 3 | 5982 | 2345 | 927 | 9058 | 3 | 20 | 22-MAR-18 | 1120 |
| 4 | 739 | 101 | 222 | 2957 | 4 | 0 | 23-DEC-17 | 1529 |
| 5 | 679 | 1111 | 101 | 3395 | 5 | 300 | 28-JAN-18 | 1748 |



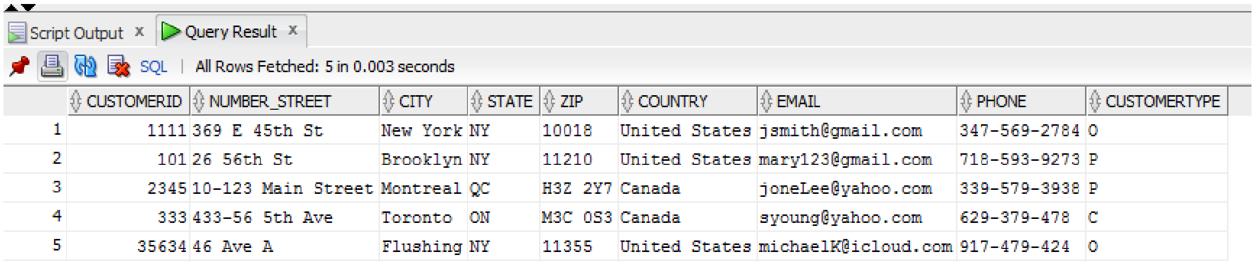
All Rows Fetched: 5 in 0.001 seconds

| | TRANSACTIONID | CUSTOMERID | LOCATIONID | ACCOUNTNO | TRANSACTIONTYPEID | TRANSACTIONAMOUNT | TRANSACTIONDATE | TRANSACTIONTIME |
|---|---|---|---|---|---|---|---|---|
| 1 | 2742 | 1111 | 101 | 9058 | 2 | 100 | 26-MAY-18 | 2053 |
| 2 | 2492 | 333 | 274 | 6903 | 1 | 0 | 12-MAR-17 | 1240 |
| 3 | 5982 | 2345 | 927 | 9058 | 3 | 20 | 22-MAR-18 | 1120 |
| 4 | 739 | 101 | 222 | 2957 | 5 | 200 | 23-DEC-17 | 1529 |
| 5 | 679 | 1111 | 101 | 3395 | 5 | 300 | 28-JAN-18 | 1748 |

<u>DELETE STATEMENT</u>

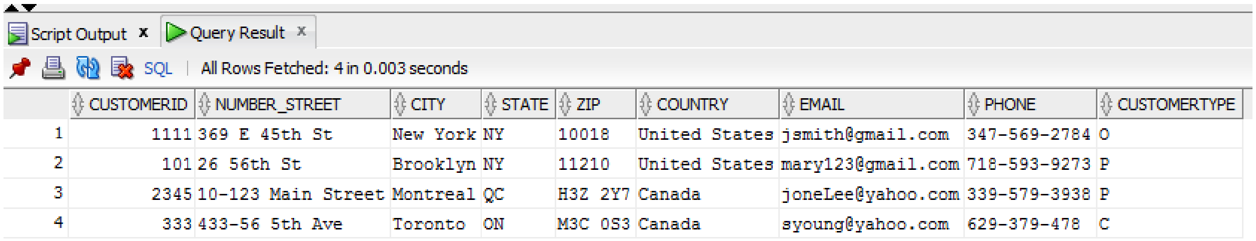
1. The following query deletes the customer with customer id 35634 from the Customer table.

```
DELETE FROM Customer
WHERE CustomerID = 35634;
```



All Rows Fetched: 5 in 0.003 seconds

| | CUSTOMERID | NUMBER_STREET | CITY | STATE | ZIP | COUNTRY | EMAIL | PHONE | CUSTOMERTYPE |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1111 | 369 E 45th St | New York | NY | 10018 | United States | jsmith@gmail.com | 347-569-2784 | O |
| 2 | 101 | 26 56th St | Brooklyn | NY | 11210 | United States | mary123@gmail.com | 718-593-9273 | P |
| 3 | 2345 | 10-123 Main Street | Montreal | QC | H3Z 2Y7 | Canada | joneLee@yahoo.com | 339-579-3938 | P |
| 4 | 333 | 433-56 5th Ave | Toronto | ON | M3C 0S3 | Canada | syoung@yahoo.com | 629-379-478 | C |
| 5 | 35634 | 46 Ave A | Flushing | NY | 11355 | United States | michaelK@icloud.com | 917-479-424 | O |

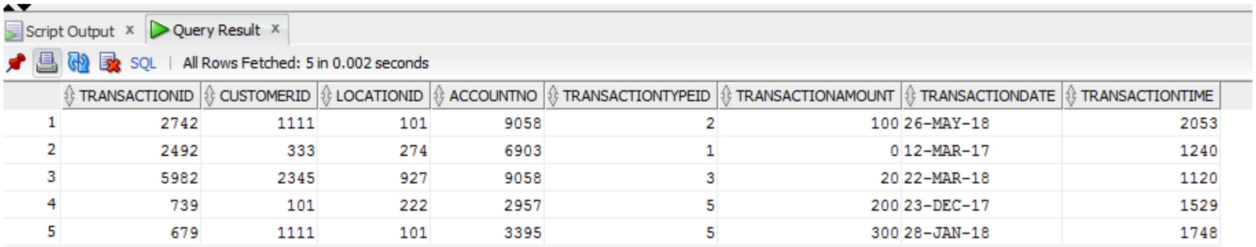


All Rows Fetched: 4 in 0.003 seconds

| | CUSTOMERID | NUMBER_STREET | CITY | STATE | ZIP | COUNTRY | EMAIL | PHONE | CUSTOMERTYPE |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1111 | 369 E 45th St | New York | NY | 10018 | United States | jsmith@gmail.com | 347-569-2784 | O |
| 2 | 101 | 26 56th St | Brooklyn | NY | 11210 | United States | mary123@gmail.com | 718-593-9273 | P |
| 3 | 2345 | 10-123 Main Street | Montreal | QC | H3Z 2Y7 | Canada | joneLee@yahoo.com | 339-579-3938 | P |
| 4 | 333 | 433-56 5th Ave | Toronto | ON | M3C 0S3 | Canada | syoung@yahoo.com | 629-379-478 | C |

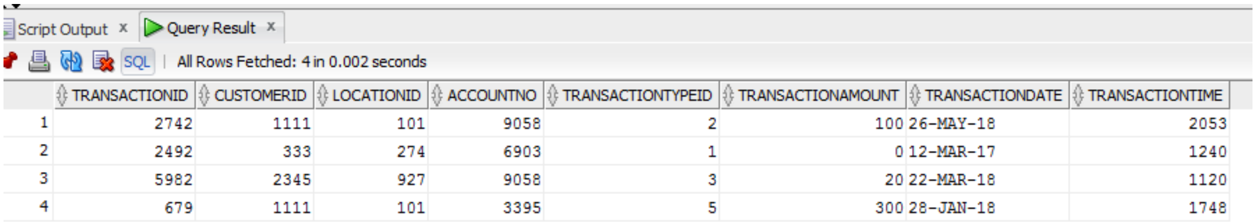
2. The following query deletes the transaction record of transaction id 739 from the Transaction table.

```
DELETE FROM Transaction
WHERE TransactionID = 739;
```



All Rows Fetched: 5 in 0.002 seconds

| | TRANSACTIONID | CUSTOMERID | LOCATIONID | ACCOUNTNO | TRANSACTIONTYPEID | TRANSACTIONAMOUNT | TRANSACTIONDATE | TRANSACTIONTIME |
|---|---|---|---|---|---|---|---|---|
| 1 | 2742 | 1111 | 101 | 9058 | 2 | 100 | 26-MAY-18 | 2053 |
| 2 | 2492 | 333 | 274 | 6903 | 1 | 0 | 12-MAR-17 | 1240 |
| 3 | 5982 | 2345 | 927 | 9058 | 3 | 20 | 22-MAR-18 | 1120 |
| 4 | 739 | 101 | 222 | 2957 | 5 | 200 | 23-DEC-17 | 1529 |
| 5 | 679 | 1111 | 101 | 3395 | 5 | 300 | 28-JAN-18 | 1748 |



All Rows Fetched: 4 in 0.002 seconds

| | TRANSACTIONID | CUSTOMERID | LOCATIONID | ACCOUNTNO | TRANSACTIONTYPEID | TRANSACTIONAMOUNT | TRANSACTIONDATE | TRANSACTIONTIME |
|---|---|---|---|---|---|---|---|---|
| 1 | 2742 | 1111 | 101 | 9058 | 2 | 100 | 26-MAY-18 | 2053 |
| 2 | 2492 | 333 | 274 | 6903 | 1 | 0 | 12-MAR-17 | 1240 |
| 3 | 5982 | 2345 | 927 | 9058 | 3 | 20 | 22-MAR-18 | 1120 |
| 4 | 679 | 1111 | 101 | 3395 | 5 | 300 | 28-JAN-18 | 1748 |