# Relational Databases with MySQL Week 6 Coding Assignment

**Points possible:** 70

|  |  |  |
| --- | --- | --- |
| Category | Criteria | % of Grade |
| Functionality | Does the code work? | 25 |
| Organization | Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear. | 25 |
| Creativity | Student solved the problems presented in the assignment using creativity and out of the box thinking. | 25 |
| Completeness | All requirements of the assignment are complete. | 25 |

**Instructions:** In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week’s assignments and push this document, with your Java project code, to the repository. Add the URL for this week’s repository to this document where instructed and submit this document to your instructor when complete.

**Coding Steps:**

This week you will be working together as a **team** to create a full CRUD application.

Your console CRUD application will need to use a database to store all the application data.

As a team, decide what you want your project to do. Get instructor approval early in the week before beginning development.

You need to have at least 3 entities.

Users should be able to interact via the console (i.e. Scanner(System.in)))

Use git to collaborate.

Everyone will be graded on their individual contributions.

**Project Name:**

**Bank Backend**

**Project Team Members:**

**My Contribution to the Project:**

My class got together and laid out a template and then did our own projects. I helped get the template together and then built my own project.

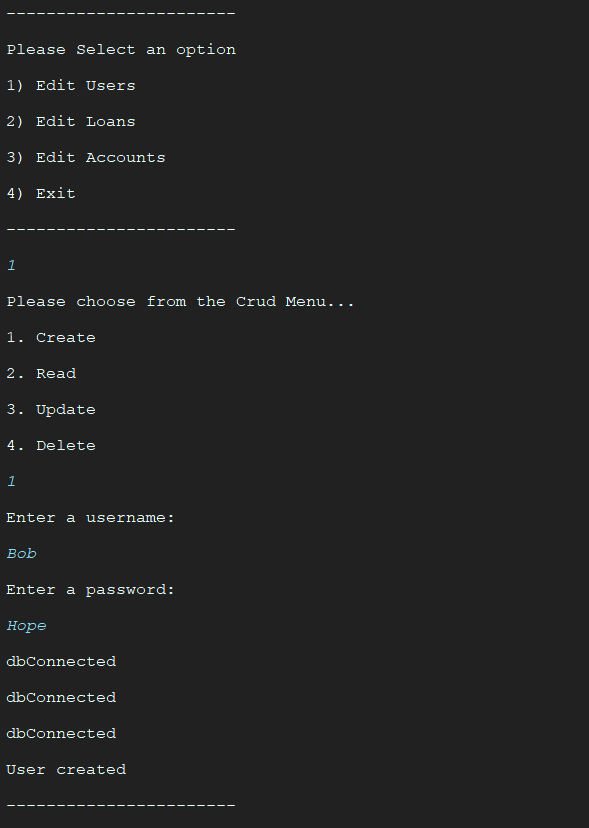
**Screenshots of Code:**

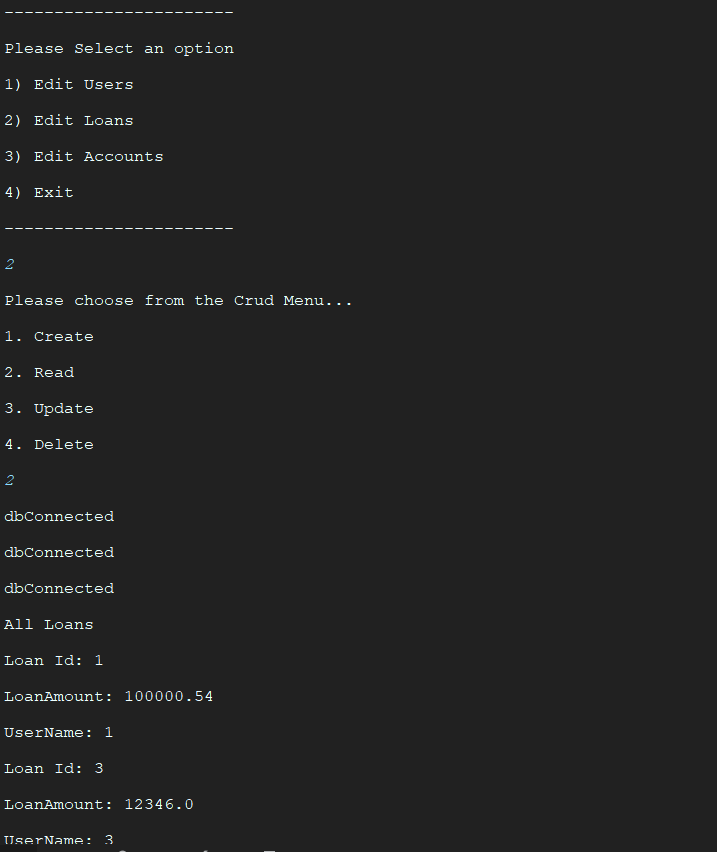
**There is a lot of code this week I will take screen shots of menu and bank services files please refer to the github repository to get all of the code.**

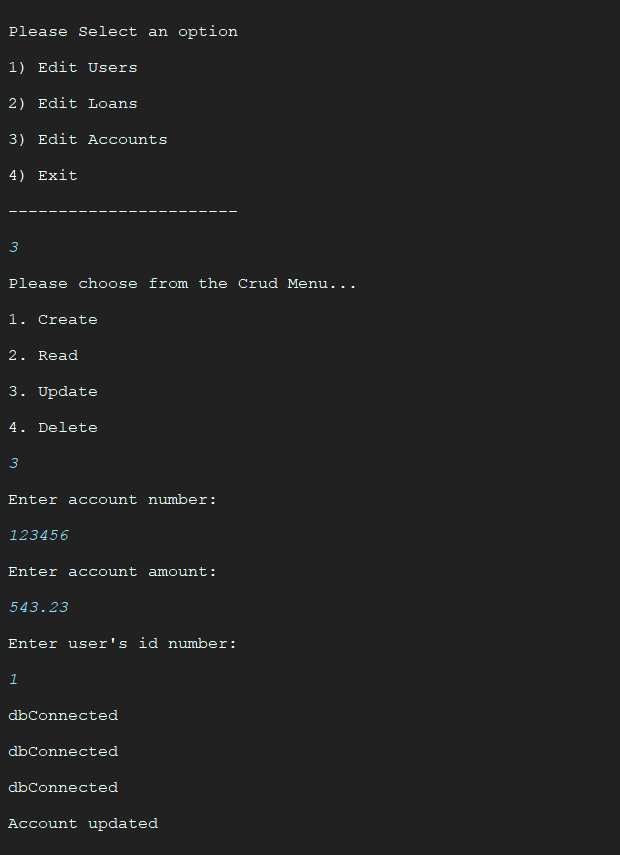
*package* com.promineo.week12.Services;  
  
*import* com.promineo.week12.Models.Account;  
*import* com.promineo.week12.Models.BankViewModel;  
*import* com.promineo.week12.Models.Loan;  
*import* com.promineo.week12.Models.User;  
  
*import* java.util.ArrayList;  
  
*public class* BankService *implements* IBankService {  
 *private* BankViewModel viewModel;  
 *private* UserService userService;  
 *private* AccountsService accountsService;  
 *private* LoanService loanService;  
  
 *public* BankService (BankViewModel model) {  
 viewModel = model;  
 userService = *new* UserService();  
 accountsService = *new* AccountsService();  
 loanService = *new* LoanService();  
 }  
  
 @Override  
 *public void* StartBankService() {  
 User user;  
 Account account;  
 Loan loan;  
  
 *switch* (viewModel.MainMenuChoice) {  
 *case* USER:  
 *switch* (viewModel.CrudChoice) {  
 *case* CREATE:  
 userService.createUser(*new* User(viewModel.UserName, viewModel.Password));  
 System.out.println("User created");  
 *break*;  
 *case* READ:  
 System.out.println("All Users");  
 ArrayList<User> users = userService.getUsers();  
 *for* (User s : users) {  
 System.out.println("User Id: " + s.getUserId());  
 System.out.println("UserName: " + s.getUserName());  
 System.out.println("Password: " + s.getPassword());  
 }  
 *break*;  
 *case* UPDATE:  
 user = userService.getUser(viewModel.UserId);  
 *if*(user != *null*) {  
 user.setUserName(viewModel.UserName);  
 user.setPassword(viewModel.Password);  
 userService.updateUser(user);  
 System.out.println("User updated");  
 } *else* {  
 System.out.println("User does not exist");  
 }  
 *break*;  
 *case* DELETE:  
 user = userService.getUser(viewModel.UserId);  
 *if* (user != *null*){  
 userService.deleteUser(user);  
 System.out.println("User was deleted");  
 } *else* {  
 System.out.println("User does not exist");  
 }  
 *break*;  
 *default*:  
 System.out.println("Invalid Choice");  
 *break*;  
 }  
 *break*;  
 *case* ACCOUNT:  
 *switch* (viewModel.CrudChoice) {  
 *case* CREATE:  
 accountsService.createAccount(*new* Account(viewModel.AccountNumber, viewModel.AccountAmount, viewModel.UserId));  
 System.out.println("Account created");  
 *break*;  
 *case* READ:  
 System.out.println("All Accounts");  
 ArrayList<Account> accounts = accountsService.getAccounts();  
 *for* (Account s : accounts) {  
 System.out.println("Account Id: " + s.getAccountNumber());  
 System.out.println("AccountAmount: " + s.getAccountAmount());  
 System.out.println("UserName: " + s.getUserId());  
 }  
 *break*;  
 *case* UPDATE:  
 account = accountsService.getAccount(viewModel.AccountNumber);  
 *if* (account != *null*) {  
 account.setAccountNumber(viewModel.AccountNumber);  
 account.setAccountAmount(viewModel.AccountAmount);  
 account.setUserId(viewModel.UserId);  
 accountsService.updateAccount(account);  
 System.out.println("Account updated");  
 } *else* {  
 System.out.println("Account does not exist");  
 }  
 *break*;  
 *case* DELETE:  
 account = accountsService.getAccount(viewModel.AccountNumber);  
 *if* (account != *null*) {  
 accountsService.deleteAccount(account);  
 System.out.println("Account was deleted");  
 } *else* {  
 System.out.println("Account does not exist");  
 }  
 *break*;  
 *default*:  
 System.out.println("Invalid Choice");  
 *break*;  
 }  
 *break*;  
 *case* LOAN:  
 *switch* (viewModel.CrudChoice) {  
 *case* CREATE:  
 loanService.createLoan(*new* Loan(viewModel.LoanAmount, viewModel.UserId));  
 System.out.println("Loan created");  
 *break*;  
 *case* READ:  
 System.out.println("All Loans");  
 ArrayList<Loan> loans = loanService.getLoans();  
 *for* (Loan s : loans) {  
 System.out.println("Loan Id: " + s.getLoanId());  
 System.out.println("LoanAmount: " + s.getLoanAmount());  
 System.out.println("UserName: " + s.getUserId());  
 }  
 *break*;  
 *case* UPDATE:  
 loan = loanService.getLoan(viewModel.LoanId);  
 *if* (loan != *null*) {  
 loan.setLoanId(viewModel.LoanId);  
 loan.setLoanAmount(viewModel.LoanAmount);  
 loan.setUserId(viewModel.UserId);  
 loanService.updateLoan(loan);  
 System.out.println("Loan updated");  
 } *else* {  
 System.out.println("Loan does not exist");  
 }  
 *break*;  
 *case* DELETE:  
 loan = loanService.getLoan(viewModel.LoanId);  
 *if* (loan != *null*) {  
 loanService.deleteLoan(loan);  
 System.out.println("Loan was deleted");  
 } *else* {  
 System.out.println("Loan does not exist");  
 }  
 *break*;  
 *default*:  
 System.out.println("Invalid Choice");  
 *break*;  
 }  
 *break*;  
 *default*:  
 System.out.println("Invalid Main Menu Choice");  
 *break*;  
 }  
 }  
}

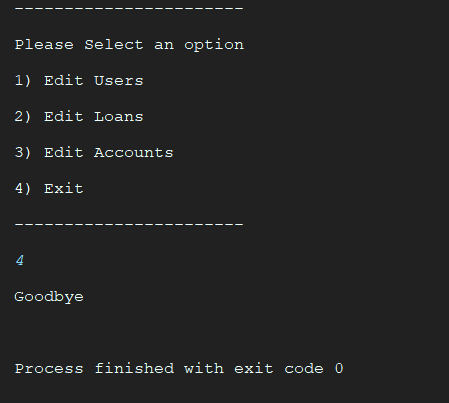
*package* com.promineo.week12.Views;  
  
*import* com.promineo.week12.Models.BankViewModel;  
*import* com.promineo.week12.Models.CrudChoice;  
*import* com.promineo.week12.Models.MainMenuChoice;  
*import* com.promineo.week12.Services.AccountsService;  
*import* com.promineo.week12.Services.BankService;  
*import* com.promineo.week12.Services.LoanService;  
*import* com.promineo.week12.Services.UserService;  
  
*import* java.util.Scanner;  
  
*public class* Menu {  
  
 *private static* BankViewModel *viewModel*;  
 *private static* Scanner *scanner* = *new* Scanner(System.in);  
  
 *public void* start() {  
 String selection = "";  
 *try* {  
 *do* {  
 *viewModel* = *new* BankViewModel();  
 *displayMainMenu*();  
 *if* (*viewModel*.MainMenuChoice != MainMenuChoice.EXIT) {  
 *displayCrudMenu*();  
 *displayDetailScreen*();  
 BankService bankService = *new* BankService(*viewModel*);  
 bankService.StartBankService();  
 }  
  
 } *while* (*viewModel*.MainMenuChoice != MainMenuChoice.EXIT);  
 System.out.println("Goodbye");  
 *scanner*.close();  
 } *catch* (Exception ex) {  
 System.out.println(ex.getMessage());  
 *scanner*.close();  
 }  
 }  
  
 *private static void* displayMainMenu() {  
 System.out.println("-----------------------");  
 System.out.println("Please Select an option");  
 System.out.println("1) Edit Users");  
 System.out.println("2) Edit Loans");  
 System.out.println("3) Edit Accounts");  
 System.out.println("4) Exit");  
 System.out.println("-----------------------");  
  
 *if* (*scanner*.hasNextInt()) {  
 *viewModel*.MainMenuChoice = MainMenuChoice.*values*()[*scanner*.nextInt() - 1];  
 }  
 }  
  
 *private static void* displayCrudMenu()  
 {  
 System.out.println("Please choose from the Crud Menu...");  
 System.out.println("1. Create");  
 System.out.println("2. Read");  
 System.out.println("3. Update");  
 System.out.println("4. Delete");  
  
 *if*(*scanner*.hasNextInt())  
 {  
 *viewModel*.CrudChoice = CrudChoice.*values*()[*scanner*.nextInt() - 1];  
 }  
 }  
 *private static void* displayDetailScreen()  
 {  
 *switch* (*viewModel*.MainMenuChoice)  
 {  
 *case* EXIT:  
  
 *break*;  
 *case* USER:  
 *switch* (*viewModel*.CrudChoice)  
 {  
 *case* CREATE:  
 System.out.println("Enter a username: ");  
 *viewModel*.UserName = *scanner*.next();  
 System.out.println("Enter a password: ");  
 viewModel.Password = scanner.next();  
 *break*;  
 *case* READ:  
 *break*;  
 *case* UPDATE:  
 System.out.println("Enter user Id: ");  
 viewModel.UserId = scanner.nextInt();  
 System.out.println("Enter username: ");  
 viewModel.UserName = scanner.next();  
 System.out.println("Enter password: ");  
 viewModel.Password = scanner.next();  
 *break*;  
 *case* DELETE:  
 System.out.println("Enter user Id: ");  
 viewModel.UserId = scanner.nextInt();  
 *break*;  
 }  
 *break*;  
 *case* ACCOUNT:  
 *switch* (*viewModel*.CrudChoice)  
 {  
 *case* CREATE:  
 System.out.println("Enter account number: ");  
 *viewModel*.AccountNumber = *scanner*.nextInt();  
 System.out.println("Enter account amount: ");  
 *viewModel*.AccountAmount = *scanner*.nextDouble();  
 System.out.println("Enter user id of account owner");  
 *viewModel*.UserId = *scanner*.nextInt();  
 *break*;  
 *case* READ:  
 *break*;  
 *case* UPDATE:  
 System.out.println("Enter account number: ");  
 *viewModel*.AccountNumber = *scanner*.nextInt();  
 System.out.println("Enter account amount: ");  
 *viewModel*.AccountAmount = *scanner*.nextDouble();  
 System.out.println("Enter user's id number: ");  
 *viewModel*.UserId = *scanner*.nextInt();  
 *break*;  
 *case* DELETE:  
 System.out.println("Enter account number to delete: ");  
 *viewModel*.AccountNumber = *scanner*.nextInt();  
 *break*;  
 }  
 *break*;  
 *case* LOAN:  
 *switch* (*viewModel*.CrudChoice)  
 {  
 *case* CREATE:  
 System.out.println("Enter loan amount: ");  
 *viewModel*.LoanAmount = *scanner*.nextDouble();  
 System.out.println("Enter the user id: ");  
 *viewModel*.UserId = *scanner*.nextInt();  
 *break*;  
 *case* READ:  
 *break*;  
 *case* UPDATE:  
 System.out.println("Enter loan Id: ");  
 viewModel.LoanId = scanner.nextInt();  
 System.out.println("Enter new loan amount: ");  
 viewModel.LoanAmount = scanner.nextDouble();  
 System.out.println("Enter user id; ");  
 viewModel.UserId = scanner.nextInt();  
 *break*;  
 *case* DELETE:  
 System.out.println("Enter loan id to delete: ");  
 viewModel.LoanId = scanner.nextInt();  
 *break*;  
 }  
 *break*;  
 }  
 }  
}

**Screenshots of Running Application:**

****

****

****

****

**URL to GitHub Repository:**