Project Code:

DB Helper File

*/\*\**

*\* build : javac -classpath jdbc.jar*

*\* run : java -classpath ".:jdbc.jar"*

*\**

*\*/*

import java.sql.\*;

public class DatabaseHelpers{

public static Connection getConnection(){

*/\*\**

*\* create connection stub*

*\* stage driver*

*\* create connection object*

*\*/*

System.*out*.println("\n\n\t--++ Connect to DB ++--");

Connection connection = null;

System.*out*.println("Attempting connection ... ");

String databasePath = "jdbc:sqlite:test.db";

String driverPath = "org.sqlite.JDBC";

try {

Class.*forName*(driverPath);

connection = DriverManager.*getConnection*(databasePath);

System.*out*.println("Database connection successful.");

} catch (Exception e) {

System.*out*.println("Could not connect to the database ... \n");

System.*err*.println( e.getClass().getName() + ": " + e.getMessage());

}

return connection;

}

public static void closeObjects(Connection connectionObject, Statement

statementObject){

System.*out*.println("\n\n\t--++ Close Result Statement and Connection " +

"Object ++--");

System.*out*.println("Attempting to close ... ");

try {

if (statementObject != null) {

statementObject.close();

System.*out*.println("Statement object closed.");

}

if (connectionObject != null) {

connectionObject.close();

System.*out*.println("Connection object closed.");

}

}

catch (Exception e) {

System.*out*.println("Could not close objects ... \n");

System.*err*.println(e.getClass().getName() + ": " + e.getMessage());

}

}

public static void executeUpdate(String update){

System.*out*.println("\n\n\t--++ Execute Update ++--");

Statement statement = null;

System.*out*.println("Attempting to execute update: " + update);

try {

Connection connection = *getConnection*();

connection.setAutoCommit(false);

statement = connection.createStatement();

statement.executeUpdate(update);

connection.commit();

*closeObjects*(connection, statement);

System.*out*.println("Update statement executed successfully.");

} catch (Exception e) {

System.*out*.println("Could not execute statement ... \n");

System.*out*.println(update);

System.*err*.println( e.getClass().getName() + ": " + e.getMessage());

}

}

public static ResultSet executeQuery(String query){

System.*out*.println("\n\n\t--++ Execute Query ++--");

Statement statement = null;

ResultSet results = null;

System.*out*.println("Attempting to execute query : \n " + query);

try {

Connection connection = *getConnection*();

connection.setAutoCommit(false);

statement = connection.createStatement();

results = statement.executeQuery(query);

System.*out*.println("Query executed successfully.");

System.*out*.println("Retrieved : " + results.getFetchSize());

results.close();

*closeObjects*(connection, statement);

} catch (Exception e) {

System.*out*.println("Could not execute query ... \n");

System.*out*.println(statement);

System.*err*.println( e.getClass().getName() + ": " + e.getMessage());

}

return results;

}

}

Menu File

*/\*\**

*\* This program is the basic menu options for the class project*

*\*/*

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.util.Scanner;

import java.util.\*;

// String databasePath = "jdbc:sqlite:test.db";

// Class.forName("org.sqlite.JDBC");

public class Menu{

public static void main(String args[]){

*displayMenu*();

}

public static void displayMenu(){

*/\*\**

*\* Displays the main menu and holds the switch statement that is*

*\* controlled by the user input.*

*\*/*

int menuChoice = 0;

int numberMenuChoices = 7;

String initChoice;

boolean token = true;

System.*out*.println("--== Stock Menu Program ==--");

System.*out*.println("1. User Options");

System.*out*.println("2. Stock Options");

System.*out*.println("3. Full User List");

System.*out*.println("4. Full Stock List");

System.*out*.println("5. Initialize Database");

System.*out*.println("6. Show Tables");

System.*out*.println("7. Exit Program");

menuChoice = *getMenuSelection*(numberMenuChoices);

switch(menuChoice){

case 1:

*userOptionMenu*();

break;

case 2:

*stockOptionMenu*();

break;

case 3:

*fullUserList*();

break;

case 4:

*fullStockList*();

break;

case 5:

System.*out*.println("This will DELETE ALL CURRENT RECORDS AND DROP ALL"

+ " TABLES. \n Do you want to continue? (Y/N)");

Scanner input = new Scanner(System.*in*);

initChoice = input.nextLine().toLowerCase();

while ( token ){

if(initChoice.equals("y")){

*initializeDatabase*();

token = false;

} else if (initChoice.equals("n")){

System.*out*.println("Returning to main menu ... ");

token = false;

} else {

System.*out*.println("Please enter Y or N ... ");

initChoice = input.nextLine().toLowerCase();

System.*out*.println("2 " + input);

}

}

break;

case 6:

*showTables*();

break;

case 7:

*quit*();

break;

}

}

public static void showTables(){

DatabaseHelpers database = new DatabaseHelpers();

ResultSet userResults = database.executeQuery("SELECT \* FROM Users");

ResultSet stocksResults = database.executeQuery("SELECT \* FROM Stocks");

ResultSet userStocksResults = database.executeQuery("SELECT \* FROM "

+ "UserStocks");

System.*out*.println("\t\t\n--== User Database ==--");

try {

//ResultSetMetaData userMeta = userResults.getMetaData();

//ResultSetMetaData stocksMeta = stocksResults.getMetaData();

//ResultSetMetaData userStockMeta = userStocksResults.getMetaData();

while (userResults.next()){

System.*out*.print("ID: " + userResults.getString("ID")

+ " FirstName: " + userResults.getString("firstName")

+ " LastName: " + userResults.getString("lastName") +

"\n");

}

} catch (Exception e) {

System.*out*.println("ERROR: ... \n");

System.*err*.println( e.getClass().getName() + ": " + e.getMessage());

}

System.*out*.println("\t\t\n--== -------- ==--");

*displayMenu*();

}

public static void userOptionMenu(){

*/\*\**

*\* Displays the user submenu and holds the switch statement that is*

*\* controlled by the user input.*

*\*/*

int menuChoice = 0;

int numberMenuChoices = 8;

System.*out*.println("--== User Menu Program ==--");

System.*out*.println("1. Get User Details");

System.*out*.println("2. Add New User");

System.*out*.println("3. Update User");

System.*out*.println("4. Delete User");

System.*out*.println("5. Get User Stocks");

System.*out*.println("6. Add User Stocks");

System.*out*.println("7. Remove User Stocks");

System.*out*.println("8. Return to Main Menu");

menuChoice = *getMenuSelection*(numberMenuChoices);

switch(menuChoice){

case 1:

System.*out*.println("User Details");

*userOptionMenu*();

break;

case 2:

System.*out*.println("Add User");

*userOptionMenu*();

break;

case 3:

System.*out*.println("Update User");

*userOptionMenu*();

break;

case 4:

System.*out*.println("Delete User");

*userOptionMenu*();

break;

case 5:

System.*out*.println("Get User Stocks");

*userOptionMenu*();

break;

case 6:

System.*out*.println("Add User Stocks");

*userOptionMenu*();

break;

case 7:

System.*out*.println("Remove User Stocks");

*userOptionMenu*();

break;

case 8:

System.*out*.println("Returning to Main Menu");

*displayMenu*();

break;

}

}

public static void stockOptionMenu(){

*/\*\**

*\* Displays the stock submenu and holds the switch statement that is*

*\* controlled by the user input.*

*\*/*

int menuChoice = 0;

int numberMenuChoices = 4;

System.*out*.println("--== Stock Menu Program ==--");

System.*out*.println("1. Get Stock Details");

System.*out*.println("2. Add New Stock");

System.*out*.println("3. Delete Stock");

System.*out*.println("4. Return to Main Menu");

menuChoice = *getMenuSelection*(numberMenuChoices);

switch(menuChoice){

case 1:

System.*out*.println("Stock Details");

*stockOptionMenu*();

break;

case 2:

System.*out*.println("Add New Stock");

*stockOptionMenu*();

break;

case 3:

System.*out*.println("Delete Stock");

*stockOptionMenu*();

break;

case 4:

System.*out*.println("Returning to Main Menu");

*displayMenu*();

break;

}

}

public static void initializeDatabase(){

*/\*\**

*\* Initialize the database*

*\*/*

DatabaseHelpers database = new DatabaseHelpers();

String dropAllTables = "DROP TABLE Users;"

+ "DROP TABLE Stocks;"

+ "DROP TABLE UserStocks;";

String createUsersTable = "CREATE TABLE Users("

+ "ID INT PRIMARY KEY NOT NULL, "

+ "firstName TEXT NOT NULL, "

+ "lastName TEXT NOT NULL)";

String createStocksTable = "CREATE TABLE Stocks "

+ "(symbol PRIMARY KEY NOT NULL,"

+ "name TEXT NOT NULL)";

String createUserStocksTable = "CREATE TABLE UserStocks "

+ "(userID PRIMARY KEY NOT NULL,"

+ "symbol TEXT NOT NULL)";

String insertUsersRecords = "INSERT INTO Users(ID, firstName, lastName)"

+ "VALUES('1','First1', 'Last1');";;

String insertStocksRecords = "INSERT INTO Stocks(symbol, name)"

+ "VALUES('AAA','AAA Company Stock');";

String insertUserStocksRecords = "INSERT INTO UserStocks(userID, "

+ "symbol) VALUES('1', 'AAA');";

database.executeUpdate(dropAllTables);

database.executeUpdate(createUsersTable);

database.executeUpdate(createStocksTable);

database.executeUpdate(createUserStocksTable);

database.executeUpdate(insertUsersRecords);

database.executeUpdate(insertStocksRecords);

database.executeUpdate(insertUserStocksRecords);

*displayMenu*();

}

public static int getMenuSelection(int maxOption){

*/\*\**

*\* Promts the user for input. If the input is not between 1 and the*

*\* max options then the user is prompted again.*

*\*/*

Scanner reader = new Scanner(System.*in*);

int selection = 0;

do{

try{

if((selection > 0) && (selection <= maxOption))

break;

else{

System.*out*.println("\nPlease make a selection from the " +

"menu: (1 - " + maxOption + ")");

String input = reader.nextLine();

selection = Integer.*parseInt*(input);

}

}catch(Exception e){

System.*out*.println("\nPlease make a selection from the menu " +

": (1 - " + maxOption + ")");

}

}

while(true);

return selection;

}

public static void fullUserList(){

System.*out*.println("User List");

*displayMenu*();

}

public static void fullStockList(){

System.*out*.println("Stock List");

*displayMenu*();

}

public static void quit(){

*/\*\**

*\* Quits the program*

*\*/*

System.*out*.println("Exiting Program ...");

System.*exit*(0);

}

}

Project in Action:



