package m6.table\_model;

import m6.ConnectionManager;

import javax.swing.\*;

import javax.swing.table.AbstractTableModel;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Vector;

public class AccountTableModel extends AbstractTableModel {

private Vector<String> columnNames = new Vector<>();

private Vector<Object[]> data;

private JFrame ui;

public AccountTableModel(JFrame ui) {

columnNames.add("Username");

columnNames.add("Account Holder");

columnNames.add("Account Number");

columnNames.add("Balance");

data = readFromDb();

this.ui = ui;

}

private Vector<Object[]> readFromDb() {

Connection conn = ConnectionManager.getInstance().getConnection();

Vector<Object[]> v = new Vector<>();

try {

PreparedStatement ps = conn.prepareStatement(

"SELECT login.name, login.username, account.accountNumber, account.balance FROM account, customer, login " +

"WHERE account.accountNumber=customer.accountNumber AND login.username=customer.username " +

"ORDER BY login.name ASC"

);

ResultSet rs = ps.executeQuery();

while (rs.next()) {

String name = rs.getString("login.name");

String username = rs.getString("login.username");

String accountNumber = rs.getString("account.accountNumber");

double balance = rs.getDouble("account.balance");

v.add(new Object[]{username, name, accountNumber, balance});

}

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(ui, "Error! Failed to fetch data.");

}

return v;

}

public int getColumnCount() {

return columnNames.size();

}

public int getRowCount() {

return data.size();

}

public String getColumnName(int col) {

return columnNames.get(col);

}

public Object getValueAt(int row, int col) {

return data.get(row)[col];

}

/\*

\* JTable uses this method to determine the default renderer/

\* editor for each cell.

\*/

public Class getColumnClass(int c) {

return getValueAt(0, c).getClass();

}

/\*

\* Don't need to implement this method unless your table's

\* editable.

\*/

public boolean isCellEditable(int row, int col) {

//Note that the data/cell address is constant,

//no matter where the cell appears onscreen.

// if (col < 2) {

// return false;

// } else {

// return true;

// }

return false;

}

/\*

\* Don't need to implement this method unless your table's

\* data can change.

\*/

public void setValueAt(Object value, int row, int col) {

data.get(row)[col] = value;

// fireTableCellUpdated(row, col);

}

private void printDebugData() {

int numRows = getRowCount();

int numCols = getColumnCount();

for (int i=0; i < numRows; i++) {

System.out.print(" row " + i + ":");

for (int j=0; j < numCols; j++) {

System.out.print(" " + data.get(i)[j]);

}

System.out.println();

}

System.out.println("--------------------------");

}

}