package edu.boisestate.cs410.charity;  
  
import com.budhash.cliche.Command;  
import com.budhash.cliche.ShellFactory;  
  
import java.io.IOException;  
import java.math.BigDecimal;  
import java.sql.\*;  
  
public class CharityShell {  
 private final Connection db;  
  
 public CharityShell(Connection cxn) {  
 db = cxn;  
 }  
  
 @Command  
 public void funds() throws SQLException {  
 String query = "SELECT fund\_id, fund\_name FROM fund";  
 try (Statement stmt = db.createStatement();  
 ResultSet rs = stmt.executeQuery(query)) {  
 System.out.format("Funds:%n");  
 while (rs.next()) {  
 System.out.format("%d: %s%n",  
 rs.getInt("fund\_id"),  
 rs.getString("fund\_name"));  
 }  
 }  
 }  
  
 @Command  
 public void donor(int id) throws SQLException {  
 String query = "SELECT donor\_name, donor\_address, donor\_city, donor\_state, donor\_zip FROM donor WHERE donor\_id = ?";  
 try (PreparedStatement stmt = db.prepareStatement(query)) {  
 stmt.setInt(1, id);  
 try (ResultSet rs = stmt.executeQuery()) {  
 if (!rs.next()) {  
 System.err.format("%d: donor does not exist%n", id);  
 return;  
 }  
 System.out.format("%s%n", rs.getString("donor\_name"));  
 System.out.format("%s%n", rs.getString("donor\_address"));  
 System.out.format("%s, %s %s%n",  
 rs.getString("donor\_city"),  
 rs.getString("donor\_state"),  
 rs.getString("donor\_zip"));  
 }  
 }  
 }  
  
 @Command  
 public void renameDonor(int id, String name) throws SQLException {  
 String query = "UPDATE donor SET donor\_name = ? WHERE donor\_id = ?";  
 try (PreparedStatement stmt = db.prepareStatement(query)) {  
 stmt.setString(1, name);  
 stmt.setInt(2, id);  
 System.out.format("Renaming donor %d to %s%n", id, name);  
 int nrows = stmt.executeUpdate();  
 System.out.format("updated %d donors%n", nrows);  
 }  
 }  
  
 @Command  
 public void addGift(int donor, String date, String... allocs) throws SQLException {  
 String insertGift = "INSERT INTO gift (donor\_id, gift\_date) VALUES (?, ?)";  
 String allocate = "INSERT INTO gift\_fund\_allocation (gift\_id, fund\_id, amount) VALUES (?, ?, ?)";  
 int giftId;  
 db.setAutoCommit(false);  
 try {  
 try (PreparedStatement stmt = db.prepareStatement(insertGift, Statement.RETURN\_GENERATED\_KEYS)) {  
 stmt.setInt(1, donor);  
 stmt.setString(2, date);  
 stmt.executeUpdate();  
 // fetch the generated gift\_id!  
 try (ResultSet rs = stmt.getGeneratedKeys()) {  
 if (!rs.next()) {  
 throw new RuntimeException("no generated keys???");  
 }  
 giftId = rs.getInt(1);  
 System.out.format("Creating gift %d%n", giftId);  
 }  
 }  
 try (PreparedStatement stmt = db.prepareStatement(allocate)) {  
 for (int i = 0; i < allocs.length; i += 2) {  
 stmt.setInt(1, giftId);  
 stmt.setInt(2, Integer.parseInt(allocs[i]));  
 stmt.setBigDecimal(3, new BigDecimal(allocs[i + 1]));  
 stmt.executeUpdate();  
 }  
 }  
 db.commit();  
 } catch (SQLException | RuntimeException e) {  
 db.rollback();  
 throw e;  
 } finally {  
 db.setAutoCommit(true);  
 }  
 }  
  
 @Command  
 public void echo(String... args) {  
 for (int i = 0; i < args.length; i++) {  
 if (i > 0) {  
 System.out.print(' ');  
 }  
 System.out.print(args[i]);  
 }  
 System.out.println();  
 }  
  
 public static void main(String[] args) throws IOException, SQLException {  
 String dbUrl = args[0];  
 try (Connection cxn = DriverManager.getConnection("jdbc:" + dbUrl)) {  
 CharityShell shell = new CharityShell(cxn);  
 ShellFactory.createConsoleShell("charity", "", shell)  
 .commandLoop();  
 }  
 }  
}