package edu.boisestate.cs410.articles;  
  
import com.budhash.cliche.Command;  
import com.budhash.cliche.ShellFactory;  
  
import java.io.IOException;  
import java.sql.\*;  
  
public class ArticleShell {  
 private final Connection db;  
  
 public ArticleShell(Connection cxn) {  
 db = cxn;  
 }  
  
 @Command  
 public void topAuthors() throws SQLException {  
 String query = "SELECT author\_id, author\_name, COUNT(article\_id) AS article\_count" +  
 " FROM author JOIN article\_author USING (author\_id)" +  
 " GROUP BY author\_id" +  
 " ORDER BY article\_count DESC LIMIT 10";  
 System.out.println("Top Authors by Publication Count:");  
 try (Statement stmt = db.createStatement();  
 ResultSet rs = stmt.executeQuery(query)) {  
 while (rs.next()) {  
 String name = rs.getString("author\_name");  
 int count = rs.getInt("article\_count");  
 System.out.format(" %s (with %d pubs)\n", name, count);  
 }  
 }  
 }  
  
 /\*\*  
 \* We can run  
 \* <pre>  
 \* conference-top-authors CHI  
 \* </pre>  
 \* @param series  
 \* @throws SQLException  
 \*/  
 @Command  
 public void conferenceTopAuthors(String series) throws SQLException {  
 String query = "SELECT author\_id, author\_name, COUNT(article\_id) AS article\_count" +  
 " FROM author JOIN article\_author USING (author\_id)" +  
 " JOIN article USING (article\_id)" +  
 " JOIN proceedings USING (proc\_id)" +  
 " JOIN conf\_series USING (cs\_id)" +  
 " WHERE cs\_hb\_key = ?" +  
 " GROUP BY author\_id" +  
 " ORDER BY article\_count DESC LIMIT 10";  
 System.out.format("Top Authors in %s by Publication Count:%n", series);  
 try (PreparedStatement stmt = db.prepareStatement(query)) {  
 // Set the first parameter (query key) to the series  
 stmt.setString(1, series);  
 // once parameters are bound we can run!  
 try (ResultSet rs = stmt.executeQuery()) {  
 while (rs.next()) {  
 String name = rs.getString("author\_name");  
 int count = rs.getInt("article\_count");  
 System.out.format(" %s (with %d pubs)\n", name, count);  
 }  
 }  
 }  
 }  
  
 /\*\*  
 \* SQL injection!  
 \* @throws SQLException  
 \*/  
 @Command  
 public void badConferenceTopAuthors(String series) throws SQLException {  
 String query = "SELECT author\_id, author\_name, COUNT(article\_id) AS article\_count" +  
 " FROM author JOIN article\_author USING (author\_id)" +  
 " JOIN article USING (article\_id)" +  
 " JOIN proceedings USING (proc\_id)" +  
 " JOIN conf\_series USING (cs\_id)" +  
 " WHERE cs\_hb\_key = '" + series + "'" +  
 " GROUP BY author\_id" +  
 " ORDER BY article\_count DESC LIMIT 10";  
 System.out.format("Top Authors in %s by Publication Count:%n", series);  
 try (Statement stmt = db.createStatement();  
 ResultSet rs = stmt.executeQuery(query)) {  
 while (rs.next()) {  
 String name = rs.getString("author\_name");  
 int count = rs.getInt("article\_count");  
 System.out.format(" %s (with %d pubs)\n", name, count);  
 }  
 }  
  
 }  
  
 public static void main(String[] args) throws IOException, SQLException {  
 String dbUrl = args[0];  
 try (Connection cxn = DriverManager.getConnection("jdbc:" + dbUrl)) {  
 ArticleShell shell = new ArticleShell(cxn);  
 ShellFactory.createConsoleShell("article", "", shell)  
 .commandLoop();  
 }  
 }  
}