**Michael Zou Wu**

**import** java.sql.\*;

**import** java.util.Scanner;

**public class** TeamDB {

**public static void** main(String[] args)

{

*// Create constants for the driver name and URL.*

*// NOTE: These values are specific for Cloudscape.*

**final** String DRIVER = **"org.apache.derby.jdbc.EmbeddedDriver"**;

**final** String DB\_URL = **"jdbc:derby:TeamDB;create=true"**;

Connection conn;

**try**

{

*// Load the JDBC driver.*

Class.*forName*(DRIVER).newInstance();

*// Create a connection to the database.*

conn = DriverManager.*getConnection*(DB\_URL);

**char** choice;

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

System.*out*.println(**"Welcome to the Sports Teams Database Manager!"**);

**do**

{

*printMenu*();

choice = keyboard.nextLine().charAt(0);

**switch** (choice)

{

**case '0'**:

*// Close the connection.*

conn.close();

**break**;

**case '1'**:

*viewTeams*(conn);

**break**;

**case '2'**:

*viewSchedule*(conn);

**break**;

**case '3'**:

*addTeams*(conn);

**break**;

**case '4'**:

*addGames*(conn);

**break**;

**case '5'**:

*enterScores*(conn);

**break**;

**case '6'**:

*beginNewSeason*(conn);

**break**;

}

}**while** (choice != **'0'**);

}

**catch**(Exception ex)

{

System.*out*.println(**"ERROR: "** + ex.getMessage());

ex.printStackTrace();

}

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// The printMenu method displays the menu choices for\**

*// the user to work with the database \**

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**public static void** printMenu()

{

System.*out*.println();

System.*out*.println(**"Select from the following options:"**);

System.*out*.println(**"1. View team standings"**);

System.*out*.println(**"2. View the schedule"**);

System.*out*.println(**"3. Add a team"**);

System.*out*.println(**"4. Add a game to the schedule"**);

System.*out*.println(**"5. Enter game scores"**);

System.*out*.println(**"6. Begin a new season"**);

System.*out*.println(**"0. Exit the program"**);

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\* Utility method to remove the tables and allow the*

*\* user to reset the database for a new season*

*\* @param conn Connection to the database*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**public static void** beginNewSeason(Connection conn)

{

**try**

{

Statement stmt = conn.createStatement();

*//remove tables if database tables have been created*

*//this will throw an exception if the tables do not exist*

stmt.execute(**"DROP TABLE Games"**);

stmt.execute(**"DROP TABLE Teams"**);

*//once the tables have been removed, call the method to*

*//create and initialize the tables*

System.*out*.println(**"Reinitializing database for a new season"**);

*createTeamDB*(conn);

}

**catch** (Exception ex)

{

*//call the method to create tables for the database*

System.*out*.println(**"Creating database for the first time"**);

*createTeamDB*(conn);

}

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\* Utility method to create the tables and initialize*

*\* the database with teams and games.*

*\* @param conn Connection to the database*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**public static void** createTeamDB(Connection conn)

{

**try**

{

Statement stmt = conn.createStatement();

*//create the table of teams*

stmt.execute(**"CREATE TABLE Teams ("** +

**"TeamName CHAR(15) NOT NULL PRIMARY KEY, "** +

**"Wins INT, "** + **"Losses INT, "** +

**"Ties INT"** + **")"**);

*//add some teams*

stmt.executeUpdate(**"INSERT INTO Teams "** +

**"(TeamName) "** +

**"VALUES ('Astros')"**);

stmt.executeUpdate(**"INSERT INTO Teams "** +

**"(TeamName) "** +

**"VALUES ('Marlins')"**);

stmt.executeUpdate(**"INSERT INTO Teams "** +

**"(TeamName) "** +

**"VALUES ('Brewers')"**);

stmt.executeUpdate(**"INSERT INTO Teams "** +

**"(TeamName) "** +

**"VALUES ('Cubs')"**);

*//create a listing of the games to be played*

stmt.execute(**"CREATE TABLE Games ("** +

**"GameNumber INT NOT NULL PRIMARY KEY, "** +

**"HomeTeam CHAR(15) NOT NULL REFERENCES Teams (TeamNumber), "** +

**"HomeTeamScore INT, "** +

**"VisitorTeam CHAR(15) NOT NULL REFERENCES Teams (TeamNumber), "** +

**"VisitorTeamScore INT"** + **")"**);

stmt.executeUpdate(**"INSERT INTO Games "** +

**"(GameNumber, HomeTeam, VisitorTeam) "** +

**"VALUES (1, 'Astros', 'Brewers')"**);

stmt.executeUpdate(**"INSERT INTO Games "** +

**"(GameNumber, HomeTeam, VisitorTeam) "** +

**"VALUES (2, 'Brewers', 'Cubs')"**);

stmt.executeUpdate(**"INSERT INTO Games "** +

**"(GameNumber, HomeTeam, VisitorTeam) "** +

**"VALUES (3, 'Cubs','Astros')"**);

}

**catch** (Exception ex)

{

System.*out*.println(**"ERROR: "** + ex.getMessage());

ex.printStackTrace();

}

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\* Allows the user to add more teams to the database*

*\* @param conn*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**public static void** addTeams(Connection conn)

{

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

**try**

{

**char** ans;

String teamName;

Statement stmt = conn.createStatement();

**do**

{

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Task 3*

*//prompt the user for a new team name*

*//write SQL statement and update the Teams table*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

System.*out*.println(**"Enter team name"**);

String tname = keyboard.nextLine();

stmt.execute(**"Insert into Teams VALUES (0,0,0);"**);

System.*out*.print(**"Do you want to enter another team: "**);

ans = keyboard.nextLine().charAt(0);

}**while** (ans == **'Y'**|| ans == **'y'**);

}

**catch** (Exception ex)

{

System.*out*.println(**"ERROR: "** + ex.getMessage());

ex.printStackTrace();

}

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\* Allows the user to add games to the schedule. A unique*

*\* game number is created for each game on the schedule.*

*\* The user will need to supply a home team name and a visitor*

*\* team name from the keyboard.*

*\* @param conn Connection to the database*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**public static void** addGames(Connection conn)

{

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

**try**

{

**char** ans;

String homeTeam;

String visitingTeam;

**int** gameNumber = 1;

Statement stmt = conn.createStatement();

*//This retrieves the retrieves the data and allows you to count*

*//the number of games already scheduled so that you add a unique game number*

String sqlStatement = **"SELECT \* from Games"**;

ResultSet result = stmt.executeQuery(sqlStatement);

**while** (result.next())

{

gameNumber++;

}

**do**

{

System.*out*.print(**"Enter the home team name: "**);

homeTeam = keyboard.nextLine();

System.*out*.print(**"Enter the visiting team number: "**);

visitingTeam = keyboard.nextLine();

sqlStatement = **"INSERT INTO Games "** +

**"(GameNumber, HomeTeam, VisitorTeam) "** + **"VALUES ('"** +

gameNumber + **"', '"** + homeTeam + **"', '"** + visitingTeam + **"')"**;

stmt.executeUpdate(sqlStatement);

System.*out*.print(**"Do you want to enter another game: "**);

ans = keyboard.nextLine().charAt(0);

}**while** (ans == **'Y'**|| ans == **'y'**);

}

**catch** (Exception ex)

{

System.*out*.println(**"ERROR: "** + ex.getMessage());

ex.printStackTrace();

}

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\* Displays a table listing the team names and season records.*

*\* Since teams have not yet played, all numbers are zero.*

*\* @param conn Connection to the database*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**public static void** viewTeams(Connection conn)

{

**try**

{

*// Create a Statement object.*

Statement stmt = conn.createStatement();

*// Create a string with a SELECT statement.*

String sqlStatement = **"SELECT \* FROM Teams"**;

*// Send the statement to the DBMS.*

ResultSet result = stmt.executeQuery(sqlStatement);

System.*out*.printf(**"%-15s s s s\n"**,

**"Team Name"**,

**"Win"**, **"Lose"**, **"Tie"**);

*// Display the contents of the result set.*

*// The result set will have 5 columns.*

**while** (result.next())

{

System.*out*.printf(**"%-15s d d d\n"**,

result.getString(**"TeamName"**),

result.getInt(**"Wins"**),

result.getInt(**"Losses"**),

result.getInt(**"Ties"**));

}

}

**catch**(Exception ex)

{

System.*out*.println(**"ERROR: "** + ex.getMessage());

}

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\* Retrieves and displays the teams and scores for all games.*

*\* @param conn Connection to the database*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**public static void** viewSchedule(Connection conn)

{

**try**

{

*// Create a Statement object.*

Statement stmt = conn.createStatement();

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// TASK 2*

*// Create a string with a SELECT statement.*

*// Send the statement to the DBMS.*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

String s = **"select GameNumber,HomeTeam,HomeTeamScore,VisitorTeam,VisitorTeamScore from Games;"**;

ResultSet rs = stmt.executeQuery(s);

*//This is a suggested column headings display*

System.*out*.println(**"List of games and scores:"**);

System.*out*.printf(**"%-6s %-20s %6s %-20s %6s\n"**,

**"GameID"**, **"Home"**, **"Score"**, **"Visitor"**, **"Score"**);

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Task 2-2c*

*// Use a while loop to display the contents of the*

*// result set.The result set will have five columns.*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

**while** (rs.next()){

System.*out*.printf(**"%-6s %-20s %6s %-20s %6s\n"**,rs.getInt(0),rs.getString(1),rs.getInt(2),

rs.getString(3),rs.getInt(4));

}

}

**catch**(Exception ex)

{

System.*out*.println(**"ERROR: "** + ex.getMessage());

}

}

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*\* Allows user to enter scores for both teams. The method will*

*\* update the Games table with the scores entered. It will*

*\* also compare the scores to determine the winning and losing*

*\* teams (or tie) and update the appropriate column in the*

*\* Teams table for each team involved in the game.*

*\* @param conn Connection to the database*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**public static void** enterScores(Connection conn)

{

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

**try**

{

**char** ans;

**int** gameNumber;

String homeTeam;

String visitingTeam;

**int** score1;

**int** score2;

String sqlStatement;

ResultSet result;

Statement stmt = conn.createStatement();

**do**

{

System.*out*.print(**"Enter the game ID: "**);

gameNumber = keyboard.nextInt();

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// TASK 4-1*

*// Get the result set from a query that selects all information*

*// for the gameNumber the user entered.*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

result = stmt.executeQuery(**"select HomeTeam,VisitorTeam from games where GameNumber="**+gameNumber);

**if**(result.next())

{

homeTeam = result.getString(**"HomeTeam"**);

visitingTeam = result.getString(**"VisitorTeam"**);

System.*out*.print(**"Enter the score for the "** + homeTeam);

score1 = keyboard.nextInt();

System.*out*.print(**"Enter the score for the "** + visitingTeam);

score2 = keyboard.nextInt();

keyboard.nextLine();

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Task 4-2*

*// Execute an update to the Games table to store the score for*

*// each team of that game number*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

stmt.executeUpdate(**"update Games set HomeTeamScore="**+score1+**",VisitorTeamScore="**+score2+**" where GameNumber="**+gameNumber);

**if** (score1 < score2)

{

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Task 4-3*

*// Retrieve the number from the appropriate column*

*// (wins, losses, or ties) for the home team, increment, and*

*// update that team's record. Do the same for the visiting team.*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

result = stmt.executeQuery(**"select \* from Teams where TeamName='"**+homeTeam+**"';"**);

**int** wins=0,loses = 0,ties=0;

**while** (result.next()){

wins = result.getInt(0);

loses = result.getInt(1);

ties = result.getInt(2);

}

loses++;

stmt.executeUpdate(**"update Teams set loses="**+loses+**" where TeamName="**+homeTeam);

result = stmt.executeQuery(**"select \* from Teams where TeamName='"**+visitingTeam+**"';"**);

wins=0;loses = 0;ties=0;

**while** (result.next()){

wins = result.getInt(0);

loses = result.getInt(1);

ties = result.getInt(2);

}

wins++;

stmt.executeUpdate(**"update Teams set wins="**+wins+**" where TeamName="**+visitingTeam);

}

**else if** (score2 < score1)

{

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Task 4-3*

*// Retrieve the number from the appropriate column*

*// (wins, losses, or ties) for the home team, increment, and*

*// update that team's record. Do the same for the visiting team.*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

result = stmt.executeQuery(**"select \* from Teams where TeamName='"**+homeTeam+**"';"**);

**int** wins=0,loses = 0,ties=0;

**while** (result.next()){

wins = result.getInt(0);

loses = result.getInt(1);

ties = result.getInt(2);

}

loses++;

stmt.executeUpdate(**"update Teams set loses="**+loses+**" where TeamName="**+homeTeam);

result = stmt.executeQuery(**"select \* from Teams where TeamName='"**+visitingTeam+**"';"**);

wins=0;loses = 0;ties=0;

**while** (result.next()){

wins = result.getInt(0);

loses = result.getInt(1);

ties = result.getInt(2);

}

wins++;

stmt.executeUpdate(**"update Teams set wins="**+wins+**" where TeamName="**+visitingTeam);

}

**else**

{

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// Task 4-3*

*// Retrieve the number from the appropriate column*

*// (wins, losses, or ties) for the home team, increment, and*

*// update that team's record. Do the same for the visiting team.*

*//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

result = stmt.executeQuery(**"select \* from Teams where TeamName='"**+homeTeam+**"';"**);

**int** wins=0,loses = 0,ties=0;

**while** (result.next()){

wins = result.getInt(0);

loses = result.getInt(1);

ties = result.getInt(2);

}

ties++;

stmt.executeUpdate(**"update Teams set ties="**+ties+**" where TeamName="**+homeTeam);

result = stmt.executeQuery(**"select \* from Teams where TeamName='"**+visitingTeam+**"';"**);

wins=0;loses = 0;ties=0;

**while** (result.next()){

wins = result.getInt(0);

loses = result.getInt(1);

ties = result.getInt(2);

}

ties++;

stmt.executeUpdate(**"update Teams set ties="**+ties+**" where TeamName="**+visitingTeam);

}

}

System.*out*.print(**"Do you want to enter another game: "**);

ans = keyboard.nextLine().charAt(0);

}**while** (ans == **'Y'**|| ans == **'y'**);

}

**catch** (Exception ex)

{

System.*out*.println(**"ERROR: "** + ex.getMessage());

ex.printStackTrace();

}

}

}