bball.java ============================================

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

\* Basketball book keeping system...

\* Objects:

\* League (Has list of Seasons, Has list of Teams)

\* Season (has list of Games)

\* Game (has 2 Teams, has GameStats)

\* GameStats

\* Team (has list of Players, has TeamStats)

\* TeamStats

\* Player (has list of PlayerStats)

\* PlayerStats

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

import java.io.\*;

import java.util.Date;

import java.util.Scanner;

import java.util.\*;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

class BBooks {

private int id;

private String name;

private ArrayList<Game> gameList = new ArrayList<Game>();

private ArrayList<Team> teamList = new ArrayList<Team>();

public BBooks(){

this.name = "";

}

public BBooks(String nm){

this.name = nm;

}

// id comes from database

public BBooks(int id, String nm){

this.id = id;

this.name = nm;

}

public String getName (){

return this.name;

}

public int getId(){

return this.id;

}

public void setGameList(ArrayList<Game> gList){

for (int j=0; j < gList.size(); j++){

this.gameList.add(gList.get(j));

}

}

public void setTeamList(ArrayList<Team> tList){

for (int j=0; j < tList.size(); j++){

this.teamList.add(tList.get(j));

}

}

}

class League {

private int id;

private String name;

private ArrayList<Game> gameList = new ArrayList<Game>();

public League(){

this.name = "";

}

public League(String nm){

this.name = nm;

}

public League(int id, String nm){

this.id = id;

this.name = nm;

}

public int getId(){

return this.id;

}

public String getName(){

return this.name;

}

}

class Game {

private String gameDate = "";

private String teamName1 = "";

private String teamName2 = "";

private int league = 0;

private int id;

private Scanner reader = new Scanner(System.in).useDelimiter("\\n");

public Game(String tname){

this.teamName1 = tname;

}

public Game(){

this.gameDate = "2017-01-12";

}

// for new - insert into database

public Game(String nm1, String nm2, String dt, int leagueID){

this.id = 0;

this.teamName1 = nm1;

this.teamName2 = nm2;

this.gameDate = dt;

this.league = leagueID;

}

// build from database read...

public Game(int id, String nm1, String nm2, String dt, int leagueID){

this.id = id;

this.teamName1 = nm1;

this.teamName2 = nm2;

this.gameDate = dt;

this.league = leagueID;

}

public String getTeamName1(){

return this.teamName1;

}

public void setTeamName1(String name){

this.teamName1 = name;

}

public String getTeamName2(){

return this.teamName2;

}

public void setTeamName2(String name){

this.teamName2 = name;

}

public String getGameDate(){

//DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

//return dateFormat.format(this.gameDate);

return this.gameDate;

}

public void setGameDate(String dt){

this.gameDate = dt;

//this.gameDate = "2017-01-12";

}

public int getLeague(){

return this.league;

}

public void setLeague(int l){

this.league = l;

}

public int getId(){

return this.id;

}

public void inputGame(){

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

String s = reader.next();

this.setTeamName1(s);

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

s = reader.next();

this.setTeamName2(s);

System.out.print("Enter the game date (yyyy-mm-dd): ");

s = reader.next();

this.setGameDate(s);

System.out.print("Enter league id: ");

int i = reader.nextInt();

this.setLeague(i);

}

}

class GameStat {

private int id = 0;

private String statName = "";

private float statValue = 0;

private int gameID = 0;

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

public GameStat (int id, String sName, float sValue, int gameID){

this.statName = sName;

this.statValue = sValue;

this.gameID = gameID;

}

public GameStat (String sName, float sValue, int gameID){

this.statName = sName;

this.statValue = sValue;

this.gameID = gameID;

}

public int getGameID(){

return this.gameID;

}

public int getId(){

return this.id;

}

public String getStatName(){

return this.statName;

}

public float getStatValue(){

return this.statValue;

}

public void display(){

System.out.println("The game statistic: (" + this.statName + ") is: " + this.statValue);

}

public void inputStat(){

System.out.print("Enter the statistic name: ");

String s = reader.next();

this.statName = s;

System.out.print("Enter the statistic value (int): ");

int i = reader.nextInt();

this.statValue = i;

}

}

class Team {

private int id = 0;

private String teamName = "";

private String school = "";

private String notes = "";

private ArrayList<Player> teamList = new ArrayList<Player>();

private ArrayList<TeamStat> teamStatsList = new ArrayList<TeamStat>();

private String[] statNames = new String[]{"Offensive Fouls", "Defensive Fouls", "Foul Shots", "Offensive Rebounds", "Defensive Rebounds"};

private Scanner reader = new Scanner(System.in).useDelimiter("\\n");

public Team (){

this.teamName = "NONAME";

}

public Team (String tName){

this.teamName = tName;

}

public Team (String tName, String school, String notes){

this.teamName = tName;

this.school = school;

this.notes = notes;

}

public Team (int id, String tName, String school, String notes){

this.id = id;

this.teamName = tName;

this.school = school;

this.notes = notes;

}

public String getName (){

return this.teamName;

}

public void setName (String name){

this.teamName = name;

}

public String getSchool(){

return this.school;

}

public void setSchool (String school){

this.school = school;

}

public String getNotes(){

return this.notes;

}

public void setNotes (String notes){

this.notes = notes;

}

public int getId(){

return this.id;

}

public void inputTeam(){

boolean r = true;

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

String s = reader.next();

this.setName(s);

System.out.print("Enter the school: ");

s = reader.next();

this.setSchool(s);

System.out.print("Enter notes: ");

s = reader.next();

this.setNotes(s);

}

public void inputStats(){

int i = 0;

for(i=0; i<this.statNames.length;i++){

TeamStat tStat = new TeamStat(this.statNames[i]);

tStat.inputStat(this.statNames[i]);

this.teamStatsList.add(tStat);

}

}

public void display(){

System.out.println("Team: " + this.teamName);

for (TeamStat ts: this.teamStatsList){

ts.display();

}

}

}

class TeamStat {

private int id = 0;

private String statName = "";

private float statValue = 0;

private int teamID = 0;

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

public TeamStat (int id, String sName, float sValue, int teamID){

this.id = id;

this.statName = sName;

this.statValue = sValue;

this.teamID = teamID;

}

public TeamStat (String sName, float sValue){

this.statName = sName;

this.statValue = sValue;

}

public TeamStat (String sName){

this.statName = sName;

this.statValue = -1;

}

public int getId(){

return this.id;

}

public void setStatValue(float val){

this.statValue = (float)val;

}

public int getTeamID(){

return this.teamID;

}

public float getStatValue(){

return this.statValue;

}

public String getStatName(){

return this.statName;

}

public void display(){

System.out.println("The team statistic: (" + this.statName + ") is: " + this.statValue);

}

public void inputStat(){

System.out.print("Enter the statistic name: ");

String s = reader.next();

this.statName = s;

System.out.print("Enter the statistic value (float): ");

float f = reader.nextFloat();

this.statValue = f;

}

public void inputStat(String name){

this.statName = name;

System.out.print("Enter the " + name + " value (int): ");

int i = reader.nextInt();

this.statValue = i;

}

}

class Player {

private int id = 0;

private String playerName = "";

private int playerNumber = 0;

private float playerHeight = 0;

private String playerNotes = "";

private int playerTeamID = 0;

private ArrayList<PlayerStat> playerStatsList = new ArrayList<PlayerStat>();

private String[] statNames = new String[]{"Offensive Fouls", "Defensive Fouls", "Foul Shots", "Offensive Rebounds", "Defensive Rebounds"};

private Scanner reader = new Scanner(System.in).useDelimiter("\\n");

public Player (){

this.playerName = "NO NAME";

}

public Player (String pName, int pNumber, float pHeight, String pNotes, int pTeamID){

this.playerName = pName;

this.playerNumber = pNumber;

this.playerHeight = pHeight;

this.playerNotes = pNotes;

this.playerTeamID = pTeamID;

}

public Player (int id, String pName, int pNumber, float pHeight, String pNotes, int pTeamID){

this.id = id;

this.playerName = pName;

this.playerNumber = pNumber;

this.playerHeight = pHeight;

this.playerNotes = pNotes;

this.playerTeamID = pTeamID;

}

public Player (String pName){

this.playerName = pName;

}

public String getPlayerNotes(){

return this.playerNotes;

}

public float getPlayerHeight(){

return this.playerHeight;

}

public int getPlayerNumber(){

return this.playerNumber;

}

public void setPlayerNumber(int i){

this.playerNumber = i;

}

public int getPlayerTeamID(){

return this.playerTeamID;

}

public void setPlayerTeamID(int i){

this.playerTeamID = i;

}

public int getId(){

return this.id;

}

public String getName (){

return this.playerName;

}

public void setName (String name){

this.playerName = name;

}

public void setPlayerHeight (float f){

this.playerHeight = f;

}

public void setPlayerNotes (String s){

this.playerNotes = s;

}

public void inputPlayer(int teamID){

System.out.print("Enter the player name: ");

String s = reader.next();

this.setName(s);

System.out.print("Enter the player number: ");

int i = reader.nextInt();

this.setPlayerNumber(i);

System.out.print("Enter the player height: ");

float f = reader.nextFloat();

this.setPlayerHeight(f);

System.out.print("Enter notes: ");

s = reader.next();

this.setPlayerNotes(s);

this.setPlayerTeamID(teamID);

}

public void inputStats(){

for(int i=0; i<this.statNames.length;i++){

PlayerStat pStat = new PlayerStat();

pStat.inputStat(this.statNames[i]);

this.playerStatsList.add(pStat);

}

}

public void display(){

System.out.println("Player: " + this.playerName);

//int i = 0;

//for (PlayerStat ps: this.playerStatsList){

// ps.display();

//}

}

}

class PlayerStat {

private int id = 0;

private int playerStatID = 0;

private int playerID = 0;

private String statName = "";

private float statValue = 0;

private int playerNumber = 0;

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

public PlayerStat (){

this.statName = "";

this.statValue = -1;

}

public PlayerStat (String sName, int sValue){

this.statName = sName;

this.statValue = (float)sValue;

}

public PlayerStat (String sName, float sValue, int playerNumber){

this.statName = sName;

this.statValue = (float)sValue;

this.playerNumber = playerNumber;

}

public PlayerStat (int id, String sName, float sValue, int playerNumber){

this.id = id;

this.statName = sName;

this.statValue = (float)sValue;

this.playerNumber = playerNumber;

}

public int getId(){

return this.id;

}

public void setStatValue(int val){

this.statValue = (float)val;

}

public int getPlayerStatID(){

return this.playerStatID;

}

public int getPlayerID(){

return this.playerID;

}

public float getStatValue(){

return this.statValue;

}

public String getStatName(){

return this.statName;

}

public void display(){

System.out.println("The player statistic: (" + this.statName + ") is: " + this.statValue);

}

public void inputStat(){

System.out.print("Enter the statistic name: ");

String s = reader.next();

this.statName = s;

System.out.print("Enter the statistic value (int): ");

int i = reader.nextInt();

this.statValue = i;

}

public void inputStat(String name){

this.statName = name;

System.out.print("Enter the " + name + " value (int): ");

int i = reader.nextInt();

this.statValue = i;

}

}

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

\* bball

\* main to create/call Books...

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

public class bball {

public static void main(String[] args){

BBooks bbooks = new BBooks();

//Team t = new Team("Galcons");

//t.inputTeam();

//Player p = new Player("Billy");

//p.inputStats();

//p.display();

//PlayerStat PS = new PlayerStat("MyStat", 6);

//PS.inputStat();

//PS.display();

}

}

=========================================================================

=========================================================================

=========================================================================

=========================================================================

=========================================================================

bballDB.java

import java.io.\*;

import java.util.Date;

import java.util.\*;

import java.sql.\*;

class DBConnect

{

Connection conn = null;

public Connection getConnection(){

if (this.conn == null){

this.openDB();

}

return this.conn;

}

public Connection openDB()

{

System.out.println("Opening DB connection");

try {

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

this.conn = DriverManager.getConnection("jdbc:sqlite:bball.db");

} catch ( Exception e ) {

System.exit(0);

}

return this.conn;

}

public void commitDB(){

try {

this.conn.commit();

} catch ( Exception e ) {

System.exit(0);

}

}

public void closeDB(){

try {

this.conn.close();

} catch ( Exception e ) {

System.exit(0);

}

}

}

public class bballDB {

public static void main(String[] args){

DBConnect DBConn = new DBConnect();

Connection conn = DBConn.openDB();

DBConn.commitDB();

DBConn.closeDB();

}

}

=========================================================================

=========================================================================

=========================================================================

=========================================================================

=========================================================================

bballDBDelete.java

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

\* Basketball book keeping system...

\* Objects:

\* DB - Database Interface (SQLite)

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

import java.io.\*;

import java.util.Date;

import java.util.\*;

import java.sql.\*;

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

\* BBooks - table

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

class DBDelete

{

public DBConnect DBConn = null;

public Connection conn = null;

DBDelete(){

this.DBConn = new DBConnect();

this.conn = DBConn.getConnection();

}

DBDelete(DBConnect DBConn){

this.DBConn = DBConn;

this.conn = DBConn.getConnection();

}

DBDelete(Connection c){

this.conn = c;

}

public void DBClose(){

this.DBConn.commitDB();

this.DBConn.closeDB();

}

public void doBooksDelete(int key)

{

Statement stmt = null;

String sql = "";

try {

stmt = this.conn.createStatement();

sql = "DELETE FROM BOOKS " +

" WHERE ID = " + key + ";";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Books");

}

}

public void doLeaguesDelete(int key)

{

Statement stmt = null;

String sql = "";

try {

stmt = this.conn.createStatement();

sql = "DELETE FROM LEAGUES " +

" WHERE ID = " + key + ";";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Leagues");

}

}

public void doGamesDelete(int key)

{

Statement stmt = null;

String sql = "";

try {

stmt = this.conn.createStatement();

sql = "DELETE FROM GAMES " +

" WHERE ID = " + key + ";";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Games");

}

}

public void doGameStatsDelete(int key)

{

Statement stmt = null;

String sql = "";

try {

stmt = this.conn.createStatement();

sql = "DELETE FROM GAME\_STATS " +

" WHERE ID = " + key + ";";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error GameStats");

}

}

public void doTeamsDelete(int key)

{

Statement stmt = null;

String sql = "";

try {

stmt = this.conn.createStatement();

sql = "DELETE FROM TEAMS " +

" WHERE ID = " + key + ";";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Teams");

}

}

public void doTeamStatsDelete(int key)

{

Statement stmt = null;

String sql = "";

try {

stmt = this.conn.createStatement();

sql = "DELETE FROM TEAM\_STATS " +

" WHERE ID = " + key + ";";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Leagues");

}

}

public void doPlayersDelete(int key)

{

Statement stmt = null;

String sql = "";

try {

stmt = this.conn.createStatement();

sql = "DELETE FROM PLAYERS " +

" WHERE ID = " + key + ";";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Players");

}

}

public void doPlayerStatsDelete(int key)

{

Statement stmt = null;

String sql = "";

try {

stmt = this.conn.createStatement();

sql = "DELETE FROM PLAYER\_STATS " +

" WHERE ID = " + key + ";";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error PlayerStats");

}

}

}

public class bballDBDelete {

public static void main(String[] args){

DBDelete DBConn = new DBDelete();

int r = 5;

DBConn.doPlayersDelete(r);

DBConn.doGamesDelete(r);

DBConn.doGameStatsDelete(r);

DBConn.doTeamsDelete(r);

DBConn.doTeamStatsDelete(r);

DBConn.doPlayerStatsDelete(r);

DBConn.doLeaguesDelete(r);

DBConn.doBooksDelete(r);

}

}

=========================================================================

=========================================================================

=========================================================================

=========================================================================

=========================================================================

bballDBTableCreate.java

import java.io.\*;

import java.util.Date;

import java.util.\*;

import java.sql.\*;

class DBTableCreate extends DBConnect

{

public void doBooksTable(int CreateDrop)

{

Statement stmt = null;

String sql = "";

String msg = "";

try {

stmt = this.conn.createStatement();

if(CreateDrop > 0){

sql = "CREATE TABLE BOOKS " +

"(ID INTEGER PRIMARY KEY AUTOINCREMENT," +

" NAME TEXT NOT NULL)";

msg = "Table Books Created";

}

else{

sql = "DROP TABLE BOOKS;";

msg = "Table Books Dropped";

}

System.out.println(msg);

stmt.executeUpdate(sql);

//stmt.close();

} catch ( Exception e ) {

System.out.println("Error Books");

System.exit(0);

}

}

public void doLeaguesTable(int CreateDrop)

{

Statement stmt = null;

String sql = "";

String msg = "";

try {

stmt = this.conn.createStatement();

if(CreateDrop > 0){

sql = "CREATE TABLE LEAGUES " +

"(ID INTEGER PRIMARY KEY AUTOINCREMENT," +

" NAME TEXT NOT NULL)";

msg = "Table Leagues Created";

}

else{

sql = "DROP TABLE LEAGUES;";

msg = "Table Books Dropped";

}

System.out.println(msg);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Leagues");

}

}

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

\* DATE - FORMAT ("yyyy-mm-dd")

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

public void doGamesTable(int CreateDrop)

{

Statement stmt = null;

String sql = "";

String msg = "";

try {

stmt = this.conn.createStatement();

if(CreateDrop > 0){

sql = "CREATE TABLE GAMES " +

"(ID INTEGER PRIMARY KEY AUTOINCREMENT," +

" TEAM\_1 TEXT NOT NULL," +

" TEAM\_2 TEXT NOT NULL," +

" GAME\_DATE DATETIME," +

" LEAGUE\_ID INT NOT NULL" +

")";

msg = "Table Games Created";

}

else{

sql = "DROP TABLE GAMES;";

msg = "Table Games Dropped";

}

System.out.println(msg);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Games");

}

}

public void doGameStatsTable(int CreateDrop)

{

Statement stmt = null;

String sql = "";

String msg = "";

try {

stmt = this.conn.createStatement();

if(CreateDrop > 0){

sql = "CREATE TABLE GAME\_STATS " +

"(ID INTEGER PRIMARY KEY AUTOINCREMENT," +

" STAT\_NAME TEXT NOT NULL," +

" STAT\_VALUE FLOAT NOT NULL," +

" GAME\_ID INT NOT NULL" +

")";

msg = "Table GameStats Created";

}

else{

sql = "DROP TABLE GAME\_STATS;";

msg = "Table Games Dropped";

}

System.out.println(msg);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error GameStats");

}

}

public void doTeamsTable(int CreateDrop)

{

Statement stmt = null;

String sql = "";

String msg = "";

try {

stmt = this.conn.createStatement();

if(CreateDrop > 0){

sql = "CREATE TABLE TEAMS " +

"(ID INTEGER PRIMARY KEY AUTOINCREMENT," +

" NAME TEXT NOT NULL," +

" SCHOOL TEXT NOT NULL," +

" NOTES TEXT" +

")";

msg = "Table Teams Created";

}

else{

sql = "DROP TABLE TEAMS;";

msg = "Table Teams Dropped";

}

System.out.println(msg);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Teams");

}

}

public void doTeamStatsTable(int CreateDrop)

{

Statement stmt = null;

String sql = "";

String msg = "";

try {

stmt = this.conn.createStatement();

if(CreateDrop > 0){

sql = "CREATE TABLE TEAM\_STATS " +

"(ID INTEGER PRIMARY KEY AUTOINCREMENT," +

" STAT\_NAME TEXT NOT NULL," +

" STAT\_VALUE FLOAT NOT NULL," +

" TEAM\_ID INT NOT NULL" +

")";

msg = "Table TeamStats Created";

}

else{

sql = "DROP TABLE TEAM\_STATS;";

msg = "Table TeamStats Dropped";

}

System.out.println(msg);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error TeamStats");

}

}

public void doPlayersTable(int CreateDrop)

{

Statement stmt = null;

String sql = "";

String msg = "";

try {

stmt = this.conn.createStatement();

if(CreateDrop > 0){

sql = "CREATE TABLE PLAYERS " +

"(ID INTEGER PRIMARY KEY AUTOINCREMENT," +

" NAME TEXT NOT NULL," +

" TEAM\_ID INT NOT NULL," +

" NUMBER INT NOT NULL," +

" HEIGHT FLOAT NOT NULL," +

" NOTES TEXT" +

")";

msg = "Table Players Created";

}

else{

msg = "Table TeamStats Dropped";

sql = "DROP TABLE PLAYERS;";

}

System.out.println(msg);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Stats");

}

}

public void doPlayerStatsTable(int CreateDrop)

{

Statement stmt = null;

String sql = "";

String msg = "";

try {

stmt = this.conn.createStatement();

if(CreateDrop > 0){

sql = "CREATE TABLE PLAYER\_STATS " +

"(ID INTEGER PRIMARY KEY AUTOINCREMENT," +

" STAT\_NAME TEXT NOT NULL," +

" STAT\_VALUE FLOAT NOT NULL," +

" PLAYER\_ID INT NOT NULL" +

")";

msg = "Table PlayerStats Created";

}

else{

sql = "DROP TABLE PLAYER\_STATS;";

msg = "Table PlayerStats Dropped";

}

System.out.println(msg);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error PlayerStats");

}

}

public void createDatabase(){

this.doBooksTable(1);

this.doLeaguesTable(1);

this.doGamesTable(1);

this.doGameStatsTable(1);

this.doTeamsTable(1);

this.doTeamStatsTable(1);

this.doPlayersTable(1);

this.doPlayerStatsTable(1);

}

public void dropDatabase(){

this.doBooksTable(0);

this.doLeaguesTable(0);

this.doGamesTable(0);

this.doGameStatsTable(0);

this.doTeamsTable(0);

this.doTeamStatsTable(0);

this.doPlayersTable(0);

this.doPlayerStatsTable(0);

}

public class bballDBTableCreate {

public static void main(String[] args){

DBTableCreate DBConn = new DBTableCreate();

Connection conn = DBConn.openDB();

//DBConn.dropDatabase();

DBConn.createDatabase();

DBConn.commitDB();

DBConn.closeDB();

}

}

=========================================================================

=========================================================================

=========================================================================

=========================================================================

=========================================================================

bballDBTransInsert.java

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

\* Basketball book keeping system...

\* Objects:

\* DBTransfer - moving data between the

\* database and the memory objects...

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

import java.io.\*;

import java.util.Date;

import java.util.\*;

import java.sql.\*;

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

\* BBooks - table

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

class DBTransInsert

{

public Connection conn = null;

public DBConnect DBConn = null;

DBTransInsert(){

this.DBConn = new DBConnect();

this.conn = DBConn.getConnection();

}

DBTransInsert(DBConnect DBConn){

this.DBConn = DBConn;

this.conn = DBConn.getConnection();

}

DBTransInsert(Connection c){

this.conn = c;

}

public DBConnect getDBConn(){

return this.DBConn;

}

public int doBooksInsert(BBooks book){

String name = book.getName();

return this.doBooksInsert(name);

}

public int doBooksInsert(String bName)

{

Statement stmt = null;

String sql = "";

int lastKey = 0;

try {

stmt = this.conn.createStatement();

sql = "INSERT INTO BOOKS " +

"(NAME) VALUES ('" + bName + "')";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.exit(0);

}

// return key of the record we just inserted

try {

ResultSet rs = stmt.executeQuery( "SELECT last\_insert\_rowid() as id;" );

rs.next();

lastKey = rs.getInt("id");

rs.close();

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Leagues (insert id)");

}

return lastKey;

}

public int doLeaguesInsert(League league){

return this.doLeaguesInsert(league.getName());

}

public int doLeaguesInsert(String name)

{

Statement stmt = null;

String sql = "";

int lastKey = 0;

try {

stmt = this.conn.createStatement();

sql = "INSERT INTO LEAGUES " +

"(NAME) VALUES ('" + name + "');";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Leagues");

}

try {

ResultSet rs = stmt.executeQuery( "SELECT last\_insert\_rowid() as id;" );

rs.next();

lastKey = rs.getInt("id");

rs.close();

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Leagues (insert id)");

}

return lastKey;

}

public int doGamesInsert(Game gm){

return this.doGamesInsert(gm.getTeamName1(), gm.getTeamName2(), gm.getGameDate(), gm.getLeague());

}

public int doGamesInsert(String team1, String team2, String dt, int league)

{

Statement stmt = null;

String sql = "";

int lastKey = 0;

try {

stmt = this.conn.createStatement();

sql = "INSERT INTO GAMES " +

"(TEAM\_1, TEAM\_2, GAME\_DATE, LEAGUE\_ID) VALUES ('" +

team1 + "','" +

team2 + "', '" + dt + "', " +

league + ");";

System.out.println(sql);

stmt.executeUpdate(sql);

//stmt.close();

} catch ( Exception e ) {

System.out.println("Error Games");

}

try {

ResultSet rs = stmt.executeQuery( "SELECT last\_insert\_rowid() as id;" );

rs.next();

lastKey = rs.getInt("id");

rs.close();

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Games (insert id)");

}

return lastKey;

}

public int doGameStatsInsert(GameStat gameStat){

return this.doGameStatsInsert(gameStat.getStatName(),

gameStat.getStatValue(),

gameStat.getGameID());

}

public int doGameStatsInsert(String statName, float statVal, int game)

{

Statement stmt = null;

String sql = "";

int lastKey = 0;

try {

stmt = this.conn.createStatement();

sql = "INSERT INTO GAME\_STATS " +

"(STAT\_NAME, STAT\_VALUE, GAME\_ID) VALUES ('" +

statName + "','" +

(float)statVal + "', " +

game + ");";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error GameStats");

}

try {

ResultSet rs = stmt.executeQuery( "SELECT last\_insert\_rowid() as id;" );

rs.next();

lastKey = rs.getInt("id");

rs.close();

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Game\_Stats (insert id)");

}

return lastKey;

}

public int doTeamsInsert(Team team){

return this.doTeamsInsert(team.getName(), team.getSchool(), team.getNotes());

}

public int doTeamsInsert(String name, String school, String notes)

{

Statement stmt = null;

String sql = "";

int lastKey = 0;

try {

stmt = this.conn.createStatement();

sql = "INSERT INTO TEAMS " +

"(NAME, SCHOOL, NOTES) VALUES ('" +

name + "','" +

school + "', '" +

notes + "');";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Teams");

}

try {

ResultSet rs = stmt.executeQuery( "SELECT last\_insert\_rowid() as id;" );

rs.next();

lastKey = rs.getInt("id");

rs.close();

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Teams (insert id)");

}

return lastKey;

}

public int doTeamStatsInsert(TeamStat ts){

return this.doTeamStatsInsert(ts.getStatName(), ts.getStatValue(), ts.getTeamID());

}

public int doTeamStatsInsert(String statName, float statValue, int teamID)

{

Statement stmt = null;

String sql = "";

int lastKey = 0;

try {

stmt = this.conn.createStatement();

sql = "INSERT INTO TEAM\_STATS " +

"(STAT\_NAME, STAT\_VALUE, TEAM\_ID) VALUES ('" +

statName + "', " + (float)statValue + "," + teamID + ");";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Leagues");

}

try {

ResultSet rs = stmt.executeQuery( "SELECT last\_insert\_rowid() as id;" );

rs.next();

lastKey = rs.getInt("id");

rs.close();

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Team\_Stats (insert id)");

}

return lastKey;

}

public int doPlayersInsert(Player player){

return this.doPlayersInsert(player.getName(), player.getPlayerTeamID(),

player.getPlayerNumber(), player.getPlayerHeight(),

player.getPlayerNotes());

}

public int doPlayersInsert(String name, int teamID, int number, float height, String txt)

{

Statement stmt = null;

String sql = "";

int lastKey = 0;

try {

stmt = this.conn.createStatement();

sql = "INSERT INTO PLAYERS " +

"(NAME, TEAM\_ID, NUMBER, HEIGHT, NOTES) VALUES ('" +

name + "', " + teamID + "," + number + "," + height + ",'" + txt + "');";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Players");

}

try {

ResultSet rs = stmt.executeQuery( "SELECT last\_insert\_rowid() as id;" );

rs.next();

lastKey = rs.getInt("id");

rs.close();

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Players (insert id)");

}

return lastKey;

}

public int doPlayerStatsInsert(PlayerStat ps){

return this.doPlayerStatsInsert(ps.getStatName(), ps.getStatValue(), ps.getPlayerID());

}

public int doPlayerStatsInsert(String statName, float statValue, int playerID)

{

Statement stmt = null;

String sql = "";

int lastKey = 0;

try {

stmt = this.conn.createStatement();

sql = "INSERT INTO PLAYER\_STATS " +

"(STAT\_NAME, STAT\_VALUE, PLAYER\_ID) VALUES ('" +

statName + "', " + (float)statValue + "," + playerID + ");";

System.out.println(sql);

stmt.executeUpdate(sql);

stmt.close();

} catch ( Exception e ) {

System.out.println("Error PlayerStats");

}

try {

ResultSet rs = stmt.executeQuery( "SELECT last\_insert\_rowid() as id;" );

rs.next();

lastKey = rs.getInt("id");

rs.close();

stmt.close();

} catch ( Exception e ) {

System.out.println("Error PlayerStats (insert id)");

}

return lastKey;

}

}

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

\* bball

\* main to create/call Books...

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

public class bballDBTransInsert {

public static void main(String[] args){

DBTransInsert DBTransInsert = new DBTransInsert();

DBDelete DBDelete = new DBDelete(DBTransInsert.getDBConn());

//Connection conn = DBTransInsert.openDB();

DBDelete.doLeaguesDelete(2);

DBDelete.doLeaguesDelete(3);

League league = new League("And a new League");

DBTransInsert.doLeaguesInsert(league);

DBDelete.doPlayersDelete(2);

DBDelete.doPlayersDelete(3);

Player player = new Player("Foo Bar", 13, (float)7.2, "The foo dude", 7);

DBTransInsert.doPlayersInsert(player);

DBDelete.doPlayerStatsDelete(2);

DBDelete.doPlayerStatsDelete(3);

PlayerStat pStat = new PlayerStat("BigStat", (float)3.4, 17);

DBTransInsert.doPlayerStatsInsert(pStat);

DBDelete.doTeamsDelete(7);

DBDelete.doTeamsDelete(8);

Team team = new Team("Bunnies", "Boston Latin", "These are ubiquitous notes");

DBTransInsert.doTeamsInsert(team);

DBDelete.doTeamStatsDelete(2);

DBDelete.doTeamStatsDelete(3);

TeamStat tStat = new TeamStat(17, "BigStat", (float)3.4, 17);

DBTransInsert.doTeamStatsInsert(tStat);

DBDelete.doGamesDelete(7);

DBDelete.doGamesDelete(8);

Game game = new Game("MyTeam", "YourTeam", "2017-08-28", 1);

DBTransInsert.doGamesInsert(game);

DBDelete.doGameStatsDelete(2);

DBDelete.doGameStatsDelete(3);

GameStat gStat = new GameStat("Big Game Stat", (float)3.9, 127);

DBTransInsert.doGameStatsInsert(gStat);

DBDelete.doBooksDelete(7);

DBDelete.doBooksDelete(8);

BBooks bbooks = new BBooks("And a new Book");

DBTransInsert.doBooksInsert(bbooks);

}

}

=========================================================================

=========================================================================

=========================================================================

=========================================================================

=========================================================================

bballDBTransSelect.java

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

\* Basketball book keeping system...

\* Objects:

\* DBTransfer - moving data between the

\* database and the memory objects...

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

import java.io.\*;

import java.util.Date;

import java.util.\*;

import java.sql.\*;

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

\* BBooks - table

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

class DBTransSelect

{

public Connection conn = null;

public DBConnect DBConn = null;

DBTransSelect(){

this.DBConn = new DBConnect();

this.conn = DBConn.getConnection();

}

DBTransSelect(DBConnect DBConn){

this.DBConn = DBConn;

this.conn = DBConn.getConnection();

}

DBTransSelect(Connection c){

this.conn = c;

}

public DBConnect getDBConn(){

return this.DBConn;

}

// giving access to DBConnect functions...

//public DBConnect getDBase(){

// return this.DBConn;

//}

public ArrayList<BBooks> doBooksSelect()

{

Statement stmt = null;

String sql = "";

ArrayList<BBooks> bbookList = new ArrayList<BBooks>();

BBooks book = null;

try {

stmt = this.conn.createStatement();

sql = "SELECT \* FROM BOOKS; ";

ResultSet rs = stmt.executeQuery(sql);

try {

while(rs.next()){

int id = rs.getInt("ID");

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

book = new BBooks(id, name);

bbookList.add(book);

}

} catch ( Exception eN ) {

System.out.println("Error RecordSet Books (cursor next)");

}

stmt.close();

} catch ( Exception eS ) {

System.out.println("Error Books (SELECT)");

}

return bbookList;

}

public ArrayList<League> doLeaguesSelect()

{

ArrayList<League> leagueList = new ArrayList<League>();

Statement stmt = null;

String sql = "";

try {

stmt = this.conn.createStatement();

sql = "SELECT \* FROM LEAGUES; ";

ResultSet rs = stmt.executeQuery( sql);

try {

while(rs.next()){

int id = rs.getInt("ID");

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

League league = new League(id, name);

leagueList.add(league);

}

} catch ( Exception eN ) {

System.out.println("Error RecordSet Books (cursor next)");

}

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Leagues (SELECT)");

}

return leagueList;

}

public ArrayList<Game> doGamesSelect()

{

Statement stmt = null;

String sql = "";

ArrayList<Game> gameList = new ArrayList<Game>();

try {

stmt = this.conn.createStatement();

sql = "SELECT ID, TEAM\_1, TEAM\_2, strftime('%Y-%m-%d', GAME\_DATE) AS G\_DATE, LEAGUE\_ID FROM GAMES; ";

ResultSet rs = stmt.executeQuery(sql);

try {

while(rs.next()){

int id = rs.getInt("ID");

String name1 = rs.getString("TEAM\_1");

String name2 = rs.getString("TEAM\_2");

String dt = rs.getString("G\_DATE");

int league = rs.getInt("LEAGUE\_ID");

Game game = new Game(id, name1, name2, dt, league);

gameList.add(game);

}

} catch ( Exception eN ) {

System.out.println("Error RecordSet Games (cursor next)");

}

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Games (SELECT)");

}

return gameList;

}

public ArrayList<GameStat> doGameStatsSelect()

{

Statement stmt = null;

String sql = "";

ResultSet rs = null;

ArrayList<GameStat> gameStatsList = new ArrayList<GameStat>();

GameStat gStat = null;

try {

stmt = this.conn.createStatement();

sql = "SELECT \* FROM GAME\_STATS ; ";

rs = stmt.executeQuery(sql);

while(rs.next()){

gStat = new GameStat( rs.getInt("ID"),

rs.getString("STAT\_NAME"),

rs.getFloat("STAT\_VALUE"),

rs.getInt("GAME\_ID"));

gameStatsList.add(gStat);

}

stmt.close();

} catch ( Exception e ) {

System.out.println("Error gameStats (SELECT)");

}

return gameStatsList;

}

public ArrayList<Team> doTeamsSelect()

{

ArrayList<Team> teamList = new ArrayList<Team>();

Statement stmt = null;

String sql = "";

try {

stmt = this.conn.createStatement();

sql = "SELECT \* FROM TEAMS; ";

ResultSet rs = stmt.executeQuery(sql);

try {

while(rs.next()){

int id = rs.getInt("ID");

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

String school = rs.getString("SCHOOL");

String notes = rs.getString("NOTES");

Team team = new Team(id, name, school, notes);

teamList.add(team);

}

} catch ( Exception eN ) {

System.out.println("Error RecordSet Teams (cursor next)");

}

stmt.close();

} catch ( Exception e ) {

System.out.println("Error Teams (SELECT)");

}

return teamList;

}

public ArrayList<TeamStat> doTeamStatsSelect()

{

Statement stmt = null;

String sql = "";

ResultSet rs = null;

ArrayList<TeamStat> teamStatsList = new ArrayList<TeamStat>();

TeamStat tStat = null;

try {

stmt = this.conn.createStatement();

sql = "SELECT \* FROM TEAM\_STATS ; ";

rs = stmt.executeQuery(sql);

while(rs.next()){

tStat = new TeamStat( rs.getInt("ID"),

rs.getString("STAT\_NAME"),

rs.getFloat("STAT\_VALUE"),

rs.getInt("TEAM\_ID"));

teamStatsList.add(tStat);

}

stmt.close();

} catch ( Exception e ) {

System.out.println("Error teamStats (SELECT)");

}

return teamStatsList;

}

public ArrayList<Player> doPlayersSelect(int teamID)

{

Statement stmt = null;

String sql = "";

ResultSet rs = null;

ArrayList<Player> playerList = new ArrayList<Player>();

Player player = null;

try {

stmt = this.conn.createStatement();

sql = "SELECT \* FROM PLAYERS WHERE TEAM\_ID = " + teamID + "; ";

rs = stmt.executeQuery(sql);

while(rs.next()){

player = new Player(rs.getInt("ID"),

rs.getString("NAME"), rs.getInt("NUMBER"),

rs.getFloat("HEIGHT"), rs.getString("NOTES"),

rs.getInt("TEAM\_ID"));

playerList.add(player);

}

stmt.close();

} catch ( Exception e ) {

System.out.println("Error PlayerStats (SELECT)");

}

return playerList;

}

public ArrayList<PlayerStat> doPlayerStatsSelect()

{

Statement stmt = null;

String sql = "";

ResultSet rs = null;

ArrayList<PlayerStat> playerStatsList = new ArrayList<PlayerStat>();

PlayerStat pStat = null;

try {

stmt = this.conn.createStatement();

sql = "SELECT \* FROM PLAYER\_STATS ; ";

rs = stmt.executeQuery(sql);

while(rs.next()){

pStat = new PlayerStat( rs.getInt("ID"),

rs.getString("STAT\_NAME"),

rs.getFloat("STAT\_VALUE"),

rs.getInt("PLAYER\_ID"));

playerStatsList.add(pStat);

}

stmt.close();

} catch ( Exception e ) {

System.out.println("Error PlayerStats (SELECT)");

}

return playerStatsList;

}

public ArrayList<PlayerStat> doPlayerStatsSelect(int playerID)

{

Statement stmt = null;

String sql = "";

ResultSet rs = null;

ArrayList<PlayerStat> playerStatsList = new ArrayList<PlayerStat>();

PlayerStat pStat = null;

try {

stmt = this.conn.createStatement();

sql = "SELECT \* FROM PLAYER\_STATS WHERE PLAYER\_ID = " + playerID + "; ";

rs = stmt.executeQuery(sql);

while(rs.next()){

pStat = new PlayerStat( rs.getInt("ID"),

rs.getString("STAT\_NAME"),

rs.getFloat("STAT\_VALUE"),

rs.getInt("PLAYER\_ID"));

playerStatsList.add(pStat);

}

stmt.close();

} catch ( Exception e ) {

System.out.println("Error PlayerStats (SELECT)");

}

return playerStatsList;

}

}

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

\* main to create/call Books...

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

public class bballDBTransSelect {

public static void main(String[] args){

ArrayList<League> leagueList = new ArrayList<League>();

DBTransSelect DBTransSelect = new DBTransSelect();

DBConnect DBConn = DBTransSelect.getDBConn();

leagueList = DBTransSelect.doLeaguesSelect();

for(int i=0; i < leagueList.size(); i++){

System.out.println(leagueList.get(i).getId() + "::" + leagueList.get(i).getName());

}

// Player

ArrayList<Player> pList = new ArrayList<Player>();

pList = DBTransSelect.doPlayersSelect(7);

for(int i=0; i < pList.size(); i++){

System.out.println(pList.get(i).getId() + "::" +

pList.get(i).getName()+ "::" +

pList.get(i).getPlayerHeight() + "::" +

pList.get(i).getPlayerTeamID());

}

// Player Stats

ArrayList<PlayerStat> psList = new ArrayList<PlayerStat>();

psList = DBTransSelect.doPlayerStatsSelect();

for(int i=0; i < psList.size(); i++){

System.out.println(psList.get(i).getId() + "::" +

psList.get(i).getStatName()+ "::" +

psList.get(i).getStatValue() + "::" +

psList.get(i).getPlayerID());

}

// Teams

ArrayList<Team> teamList = new ArrayList<Team>();

teamList = DBTransSelect.doTeamsSelect();

for(int i=0; i < teamList.size(); i++){

System.out.println(teamList.get(i).getId() + "::" +

teamList.get(i).getName()+ "::" +

teamList.get(i).getSchool() + "::" +

teamList.get(i).getNotes());

}

// Player Stats

ArrayList<TeamStat> tList = new ArrayList<TeamStat>();

tList = DBTransSelect.doTeamStatsSelect();

for(int i=0; i < tList.size(); i++){

System.out.println(tList.get(i).getId() + "::" +

tList.get(i).getStatName()+ "::" +

tList.get(i).getStatValue() + "::" +

tList.get(i).getTeamID());

}

// Games

ArrayList<Game> gameList = new ArrayList<Game>();

gameList = DBTransSelect.doGamesSelect();

for(int i=0; i < gameList.size(); i++){

System.out.println(gameList.get(i).getId() + "::" +

gameList.get(i).getTeamName1()+ "::" +

gameList.get(i).getTeamName2() + "::" +

gameList.get(i).getGameDate());

}

// GameStats

System.out.println("Game Stats...");

ArrayList<GameStat> gameStatList = new ArrayList<GameStat>();

gameStatList = DBTransSelect.doGameStatsSelect();

for(int i=0; i < gameStatList.size(); i++){

System.out.println(gameStatList.get(i).getId() + "::" +

gameStatList.get(i).getStatName()+ "::" +

gameStatList.get(i).getStatValue() + "::" +

gameStatList.get(i).getGameID());

}

// BBooks

System.out.println("Book Stats...");

ArrayList<BBooks> bbookList = new ArrayList<BBooks>();

bbookList = DBTransSelect.doBooksSelect();

for(int i=0; i < bbookList.size(); i++){

System.out.println(bbookList.get(i).getId() + "::" + bbookList.get(i).getName());

}

}

}

=========================================================================

=========================================================================

=========================================================================

=========================================================================

=========================================================================