package com.zybooks.gacharpg;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.provider.BaseColumns;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.ArrayList;

public class GachaDB extends SQLiteOpenHelper {

private static final String DATABASE\_NAME = "roster.db";

private static final int VERSION = 1;

private static GachaDB sInstance;

private static Context ctx;

public static synchronized GachaDB getInstance(Context context) {

if(sInstance == null)

sInstance = new GachaDB(context.getApplicationContext());

return sInstance;

}

protected GachaDB(Context context){

super(context, DATABASE\_NAME, null, VERSION);

ctx = context;

}

private static final class RosterTable implements BaseColumns {

private static final String TABLE = "roster";

private static final String COL\_ID = "\_id";

private static final String COL\_NAME = "name";

private static final String COL\_ELEMENT = "element";

private static final String COL\_ROLE = "role";

private static final String COL\_LEVEL = "level";

private static final String COL\_UNLOCKED = "unlocked";

private static final String COL\_PARTY1 = "party1";

private static final String COL\_PARTY2 = "party2";

private static final String COL\_PARTY3 = "party3";

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("create table " + RosterTable.TABLE + " (" +

RosterTable.COL\_ID + " integer primary key autoincrement, " +

RosterTable.COL\_NAME + " text unique, " +

RosterTable.COL\_ELEMENT + " text, " +

RosterTable.COL\_ROLE + " text, " +

RosterTable.COL\_LEVEL + " text, " +

RosterTable.COL\_UNLOCKED + " text, " +

RosterTable.COL\_PARTY1 + " text, " +

RosterTable.COL\_PARTY2 + " text, " +

RosterTable.COL\_PARTY3 + " text)");

try {

BufferedReader in = new BufferedReader(new InputStreamReader(ctx.getAssets().open("default\_roster.csv")));

String reader = "";

while ((reader = in.readLine()) != null) {

String[] data = reader.split(",");

String name = data[0];

String element = data[1];

String role = data[2];

String level = data[3];

String unlocked = data[4];

ContentValues values = new ContentValues();

values.put(RosterTable.COL\_NAME, name);

values.put(RosterTable.COL\_ELEMENT, element);

values.put(RosterTable.COL\_ROLE, role);

values.put(RosterTable.COL\_LEVEL, level);

values.put(RosterTable.COL\_UNLOCKED, unlocked);

db.insert(RosterTable.TABLE, null, values);

}

in.close();

} catch(Exception e) {

//drop it like it's hot

}

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("drop table if exists " + RosterTable.TABLE);

onCreate(db);

}

public void resetDB() {

SQLiteDatabase db = getWritableDatabase();

db.execSQL("drop table if exists " + RosterTable.TABLE);

onCreate(db);

}

public String[] getAll() {

SQLiteDatabase db = getReadableDatabase();

ArrayList<String> sb = new ArrayList<>();

String projection[] = { RosterTable.COL\_NAME };

Cursor cursor = db.query(RosterTable.TABLE, projection, null, null, null, null, null);

if(cursor.moveToFirst()) {

do {

String name = cursor.getString(0);

sb.add(name);

}while (cursor.moveToNext());

}

cursor.close();

return sb.toArray(new String[sb.size()]);

}

public String[] getNames() {

SQLiteDatabase db = getReadableDatabase();

ArrayList<String> sb = new ArrayList<>();

String projection[] = { RosterTable.COL\_NAME };

String selection = RosterTable.COL\_UNLOCKED + "=?";

String[] selectionArgs = { "y" };

Cursor cursor = db.query(RosterTable.TABLE, projection, selection, selectionArgs, null, null, null);

if(cursor.moveToFirst()) {

do {

String name = cursor.getString(0);

sb.add(name);

}while (cursor.moveToNext());

}

cursor.close();

return sb.toArray(new String[sb.size()]);

}

public String[] getStats(String name) {

SQLiteDatabase db = getReadableDatabase();

String[] sb = new String[4];

String projection[] = {

RosterTable.COL\_ID,

RosterTable.COL\_ELEMENT,

RosterTable.COL\_ROLE,

RosterTable.COL\_LEVEL

};

String selection = RosterTable.COL\_NAME + " =?";

String[] selectionArgs = { name };

Cursor cursor = db.query(RosterTable.TABLE,

projection,

selection,

selectionArgs,

null,

null,

null);

if(cursor.moveToFirst()) {

do {

String id = String.valueOf(cursor.getLong(0));

String element = cursor.getString(1);

String role = cursor.getString(2);

String level = cursor.getString(3);

sb[0] = id;

sb[1] = element;

sb[2] = role;

sb[3] = level;

}while (cursor.moveToNext());

}

cursor.close();

return sb;

}

public void unlockCharacter(String name) {

SQLiteDatabase db = getWritableDatabase();

ContentValues values = new ContentValues();

values.put(RosterTable.COL\_UNLOCKED, "y");

db.update(RosterTable.TABLE, values, RosterTable.COL\_NAME + " =?", new String[] { name });

}

}