# Запити зі зв'язаних таблиць до бази даних SQLite в Android,

Як за допомогою методів query і rawQuery SQLite виконувати запити для зв'язаних таблиць. Створіть простий додаток, який буде робити запит із двох таблиць і виводити результат на екран. Таблиці буде дві - people і position. У першу (people) запишіть список людей, у другу (position) – список посад. І для кожної людини з таблиці людей буде прописано id посади з таблиці position.

package …

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

public class MainActivity extends AppCompatActivity {

final String LOG\_TAG = "myLogs";

int[] position\_id = {1, 2, 3, 4};

String[] position\_name = {"Директор", "Програміст", "Бухгалтер", "Охоронець"};

int[] position\_salary = {80000, 60000, 40000, 20000};

String[] people\_name = {"Максим", "Сергій", "Руслан", "Наталія", "Іван", "Марія", "Світлана", "Григорій"};

int[] people\_posid = {2, 3, 2, 2, 3, 1, 2, 4};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

DbHelper dbHelper = new DbHelper(this);

SQLiteDatabase sqLiteDatabase = dbHelper.getWritableDatabase();

Cursor cursor;

Log.d(LOG\_TAG, "---Table position---");

cursor = sqLiteDatabase.query("position", null, null, null, null, null, null);

logCursor(cursor);

cursor.close();

Log.d(LOG\_TAG, "--- ---");

Log.d(LOG\_TAG, "---Table people---");

cursor = sqLiteDatabase.query("people", null, null, null, null, null, null);

logCursor(cursor);

cursor.close();

Log.d(LOG\_TAG, "--- ---");

Log.d(LOG\_TAG, "---INNER JOIN with rawQuery---");

String sqlQuery = "select PL.name як Name, PS.name як Position, salary as Salary "

+ "From people as PL "

+ "inner join position as PS"

+ "on PL.posid = PS.id"

+ "where salary >?";

cursor = sqLiteDatabase.rawQuery(sqlQuery, new String[] {"40000"});

logCursor(cursor);

cursor.close();

Log.d(LOG\_TAG, "--- ---");

Log.d(LOG\_TAG, "---INNER JOIN with query---");

String table = " people as PL в join position as PS on PL.posid = PS.id";

String[] columns = {"PL.name as Name", "PS.name as Position", "salary as Salary"};

String selection = "salary <?";

String[] selectionArgs = {"40000"};

cursor = sqLiteDatabase.query(table, columns, selection, selectionArgs, null, null, null);

logCursor(cursor);

cursor.close();

Log.d(LOG\_TAG, "--- ---");

dbHelper.close();

}

void logCursor(Cursor cursor) {

if (cursor! = null) {

if (cursor.moveToFirst()) {

String str;

do {

str = "";

for (String cn : cursor.getColumnNames()) {

str = str.concat(cn + " = " + cursor.getString(cursor.getColumnIndex(cn)) + "; ");

}

Log.d(LOG\_TAG, str);

} while (cursor.moveToNext());

}

} else Log.d(LOG\_TAG, "Cursor is null");

}

}

class DbHelper extends SQLiteOpenHelper {

public DbHelper(Context context) {

super(context, "myDb", null, 1);

}

@Override

public void onCreate(SQLiteDatabase sqLiteDatabase) {

Log.d(LOG\_TAG, "---onCreate database---");

ContentValues contentValues = new ContentValues();

sqLiteDatabase.execSQL("create table position ("

+ "id integer primary key,"

+ "name text, " + "salary integer"

+ ");");

for (int i = 0; i < position\_id.length; i++) {

contentValues.clear();

contentValues.put("id", position\_id[i]);

contentValues.put("name", position\_name[i]);

contentValues.put("salary", position\_salary[i]);

sqLiteDatabase.insert("position", null, contentValues);

}

sqLiteDatabase.execSQL("create table people ("

+ "id integer primary key autoincrement, "

+ "name text,"

+ "posid integer"

+ ");");

for (int i = 0; i < people\_name.length; i++) {

contentValues.clear();

contentValues.put("name", people\_name[i]);

contentValues.put("posid", people\_posid[i]);

sqLiteDatabase.insert("people", null, contentValues);

}

}

@Override

public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

}

}