Hướng dẫn đồ án

Lưu dữ liệu vào thư mục assets, khi load chương trình lên database sẽ được nạp vào bộ nhớ của chương trình.

# Tạo database

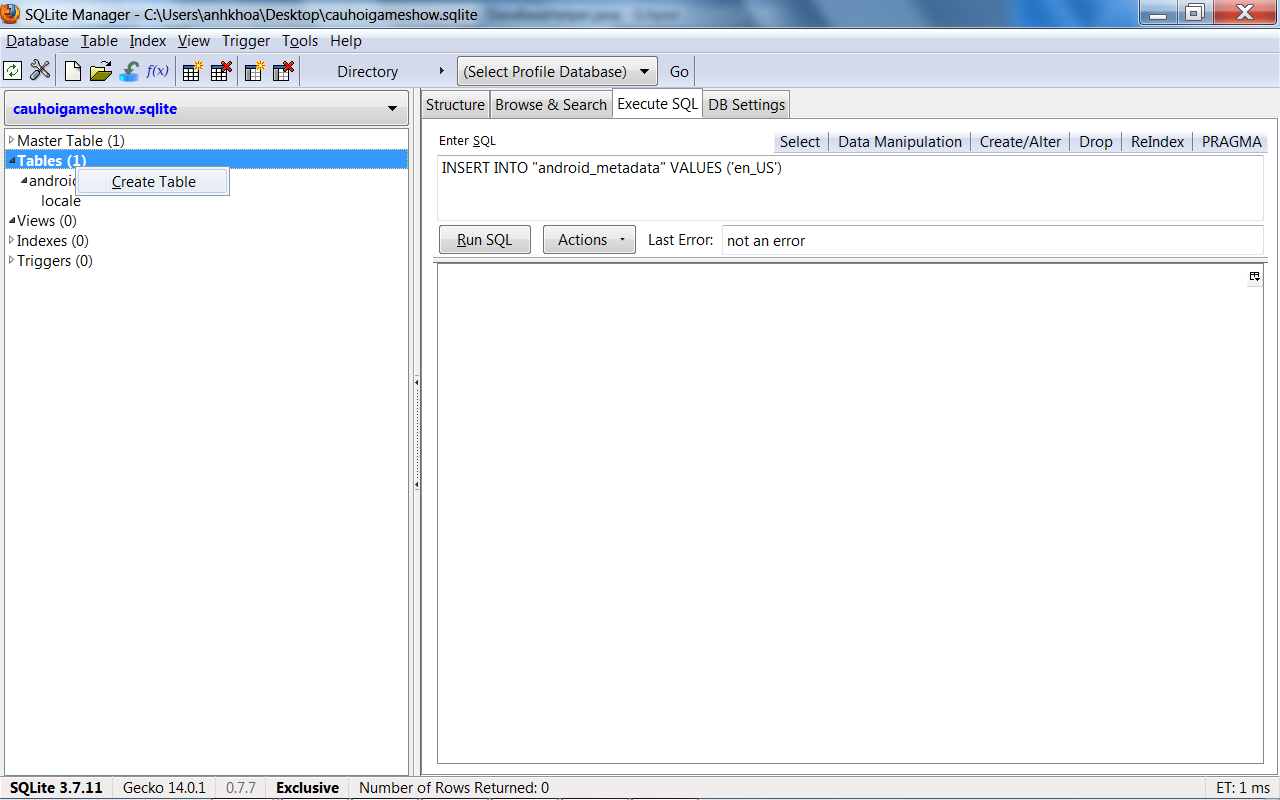
1. Mở SQLite Manager (add-on của firefox). Chọn Database -> New Database. Đặt tên database vd: “databasecauhoi” rồi chọn OK và chỉ thư mục mà database sẽ lưu trữ vd:Desktop rồi OK.
2. Trên giao diện, chọn tab “Execute SQL” và paste câu lệnh bên dưới vào rồi click nút “run SQL”.

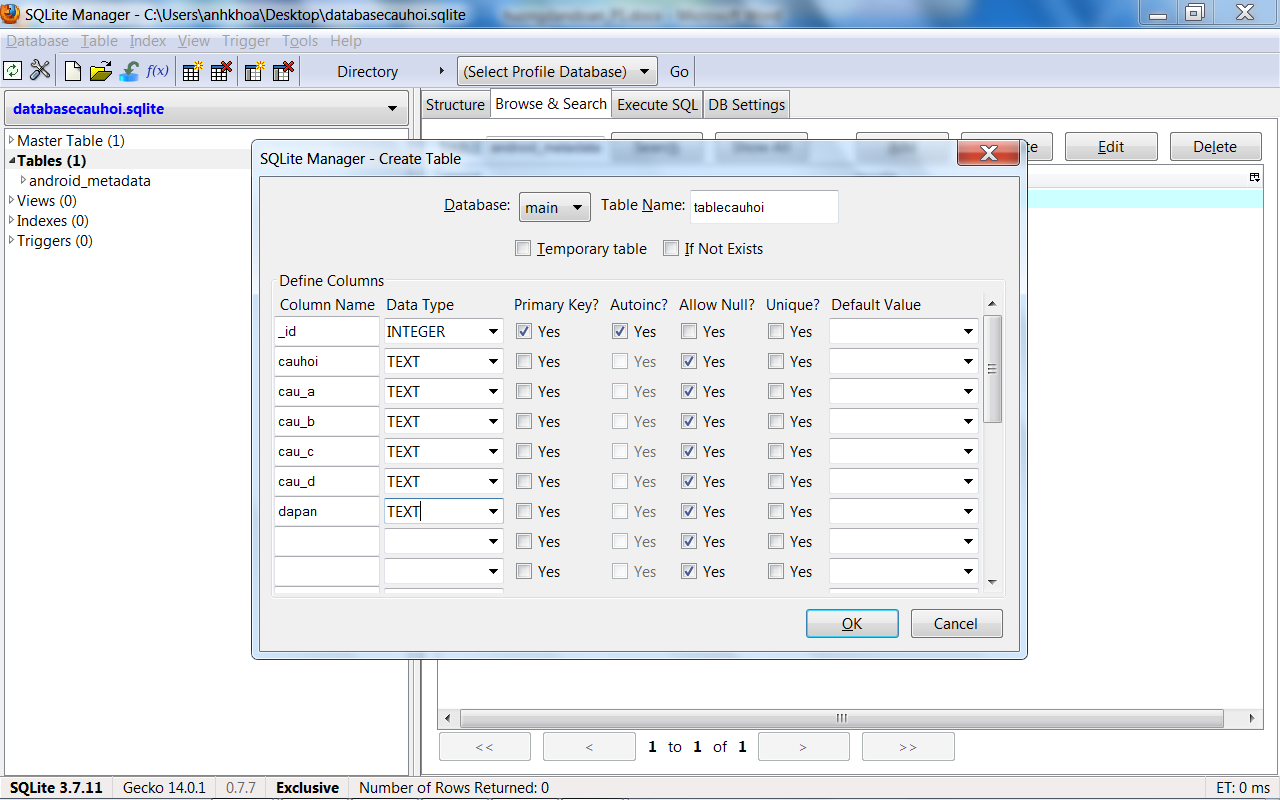
**CREATETABLE**"android\_metadata"("locale" TEXT **DEFAULT**'en\_US')

1. Chạy tiếp câu lệnh (dùng cho android phải có 1 bảng và giá trị như thế):

**INSERTINTO**"android\_metadata"**VALUES**('en\_US')

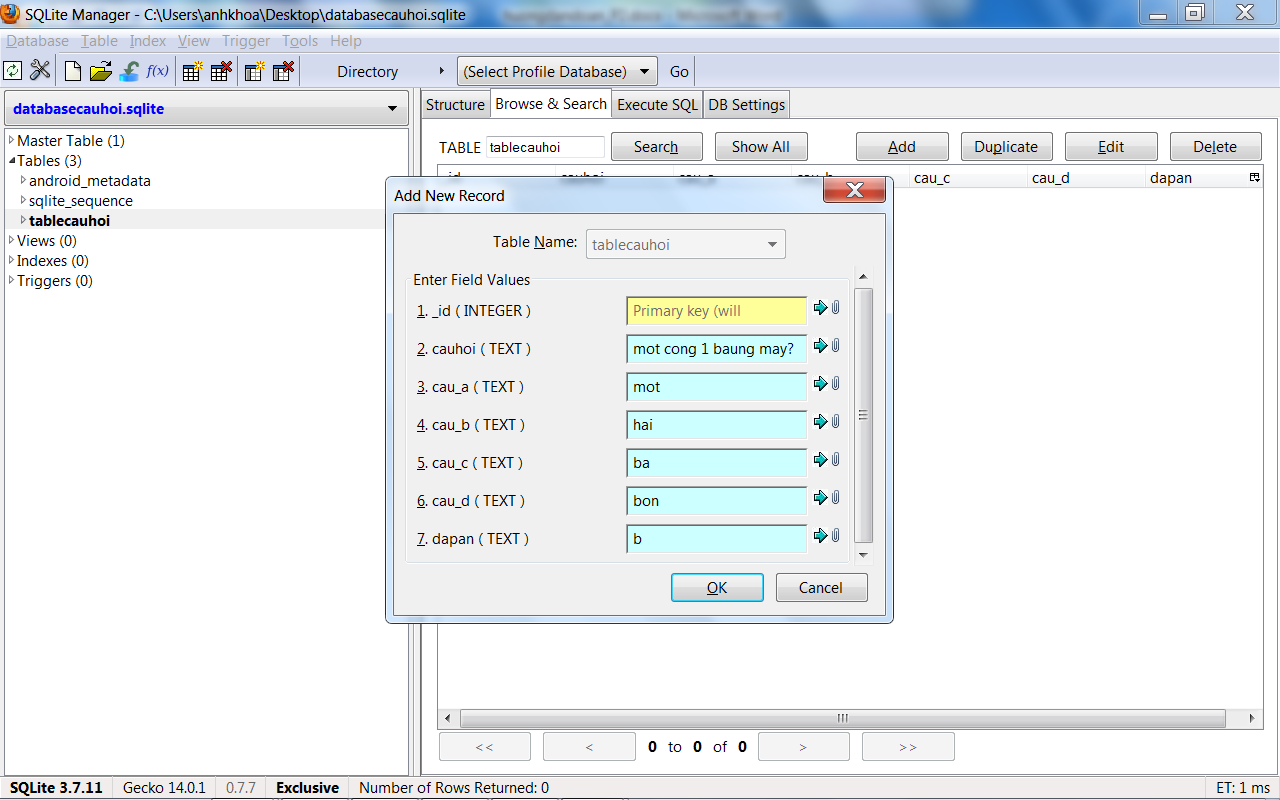
1. Click phải mục “Table” chọn “Create Table”.



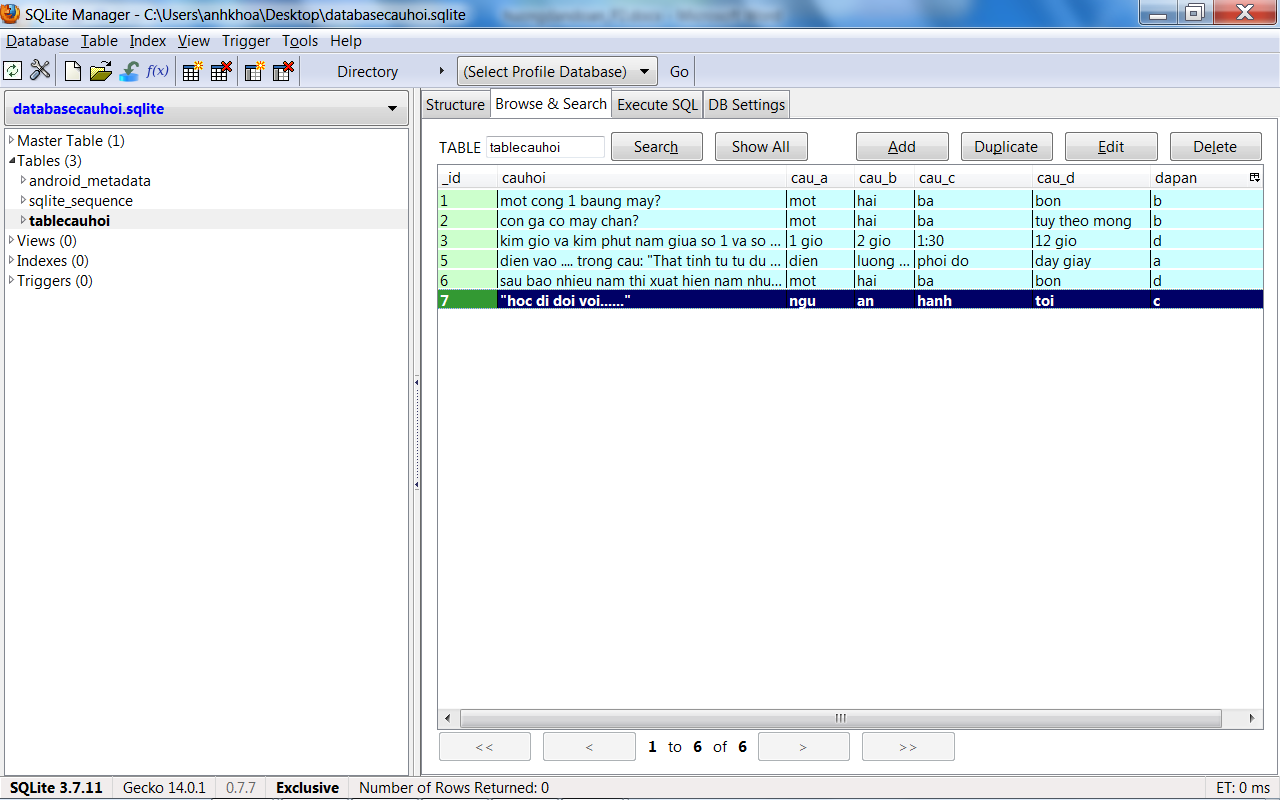
1. Nhập dữ liệu như hình.

Chú ý: tên table “tablecauhoi” , trường đầu tiên là khóa chính nên tên phải là “\_id”, có dấu “\_” trước chữ “id”, và kiểu là Integer, check vào Primay key và Autonic. Các trường còn lại kiểu text.

1. Chọn Table “tablecauhoi” rồi chọn tab “Browse & Search” rồi đến nút Add.
2. Nhập dữ liệu như hình rồi nhấn OK (nhớ id không nhập vì tự tăng).



1. Thực hiện tương tự để nhập một số câu hỏi như sau:

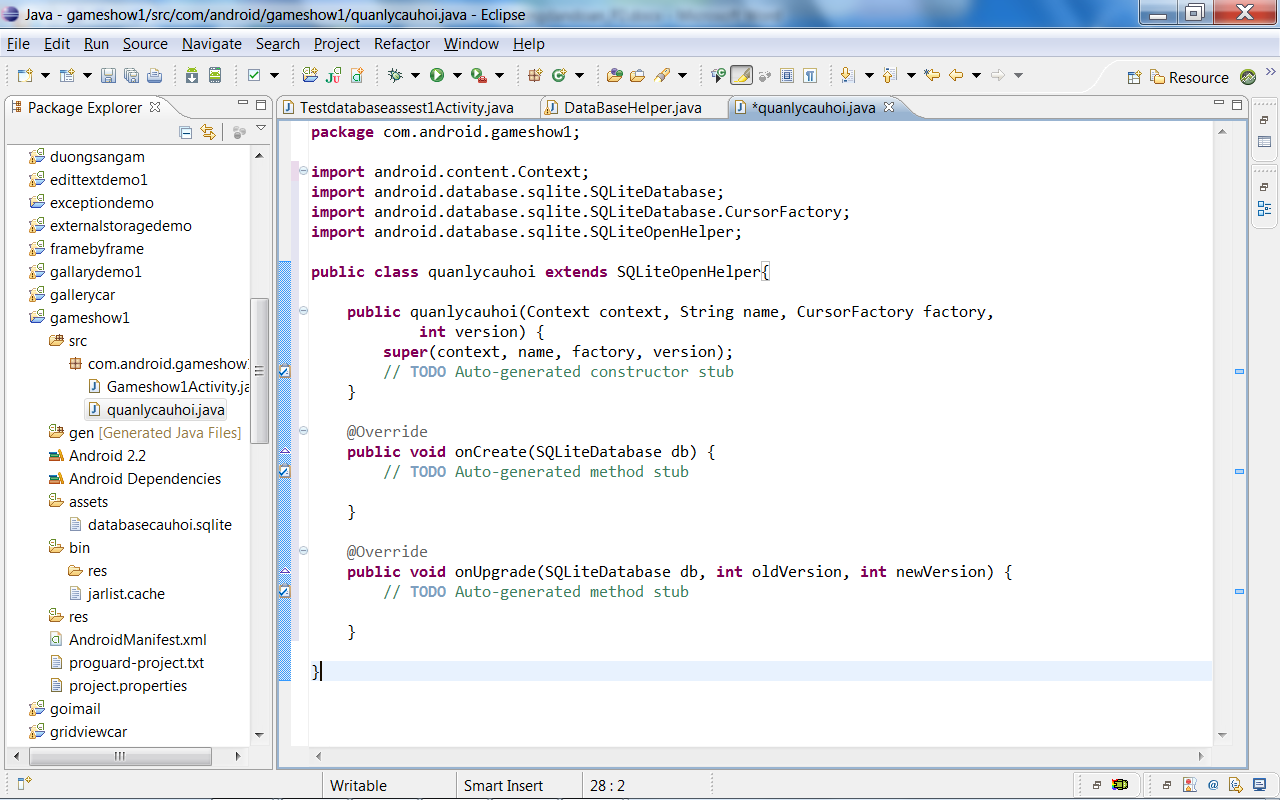


# Tạo project và đưa database vào.

1. Tạo 1 project mới tên vd: “gameshow1”.
2. Kéo database vừa tạo vào thư mục “assets” trong project.

# Tạo class để quản lý câu hỏi (chép database từ thư mục assets sang bộ nhớ của chương trình)

1. Tạo 1 class mới tên “quanlycauhoi” kế thừa từ lớp “SQLiteOpenHelper”, xây hàm tạo và override 2 phương thức onCreate, onUpgrade.



1. Khai báo các biến toàn cục:

**privatestatic** String *DB\_PATH* = "/data/data/com.android.testdatabaseassest1/databases/";

**privatestatic** String *DB\_NAME* = "databasecauhoi.sqlite";

**privatestaticfinalint***DATABASE\_VERSION*=1;

**privatestaticfinal** String *TABLE\_NAME*="tablecauhoi";

**privatestaticfinal** String *KEY\_ID*="\_id";

**privatestaticfinal** String *KEY\_CAUHOI*="cauhoi";

**privatestaticfinal** String *KEY\_A*="cau\_a";

**privatestaticfinal** String *KEY\_B*="cau\_b";

**privatestaticfinal** String *KEY\_C*="cau\_c";

**privatestaticfinal** String *KEY\_D*="cau\_d";

**privatestaticfinal** String *KEY\_DA*="dapan";

**private** SQLiteDatabase myDataBase;

**privatefinal** Context myContext;

Chú ý: phần DB\_PATH chứa đường dẫn dẫn đến database sẽ lưu vào bộ nhớ.

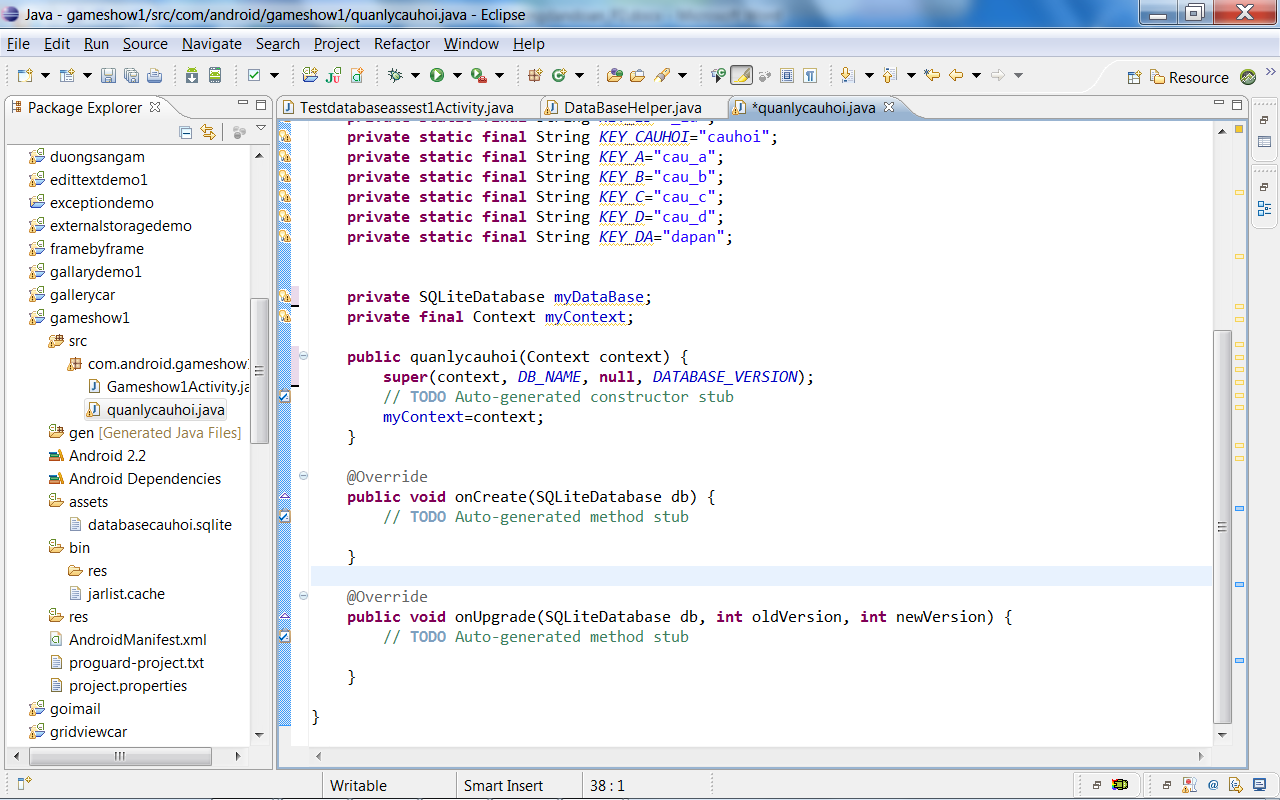
"/data/data/<tênpackage>/databases/"

Bạn phải thay phần <tênpackage> bằng tên package của project.

Phần DB\_NAME chính là tên của file database mà bạn lưu trong thư mục assect

*DB\_NAME*= "tênfiledatabase";

1. Sửa lại hàm tạo cho đơn giản hơn:



1. Tiếp theo xây dựng hàm mở và đóng database. Hàm mở database phải ném vào lỗi để bẩy lỗi khi không mở được database. Hàm này sẽ mở database lưu trong thư mục assets và gán cho biến myDataBase.

**publicvoid** openDataBase() **throws** SQLException{

//Open the database

String myPath = *DB\_PATH* + *DB\_NAME*;

myDataBase = SQLiteDatabase.*openDatabase*(myPath, **null**, SQLiteDatabase.*OPEN\_READONLY*);

}

@Override

**publicsynchronizedvoid** close() {

**if**(myDataBase != **null**)

myDataBase.close();

**super**.close();

}

1. Xây dựng tiếp 1 hàm tên “checkDataBase” hàm này có chức năng mở thử database xem mở được không và trả về 1 boolean để báo xem database đã sẵn sàng chưa.

**privateboolean** checkDataBase(){

SQLiteDatabase checkDB = **null**;

**try**{

String myPath = *DB\_PATH* + *DB\_NAME*;

checkDB = SQLiteDatabase.*openDatabase*(myPath, **null**, SQLiteDatabase.*OPEN\_READONLY*);

}**catch**(SQLiteException e){

//database chua ton tai

}

**if**(checkDB != **null**)

checkDB.close();

**return** checkDB != **null** ? **true** :**false**;

}

1. Xây hàm copyDataBase để copy dữ liệu từ file database trong thư mục assets sang file database trên bộ nhớ.

**privatevoid**copyDataBase() **throws** IOException{

//mo db trong thu muc assets nhu mot input stream

InputStream myInput = myContext.getAssets().open(*DB\_NAME*);

//duong dan den db se tao

String outFileName = *DB\_PATH* + *DB\_NAME*;

//mo db giong nhu mot output stream

OutputStream myOutput = **new**FileOutputStream(outFileName);

//truyen du lieu tu inputfile sang outputfile

**byte**[] buffer = **newbyte**[1024];

**int** length;

**while** ((length = myInput.read(buffer))>0)

{

myOutput.write(buffer, 0, length);

}

//Dong luon

myOutput.flush();

myOutput.close();

myInput.close();

}

1. Xây tiếp hàm createDataBase hàm này kiểm tra nếu database đã tồn tại thì không làm gì hết, ngược lại chưa có thì gọi hàm copydatabase để chép dữ liệu.

**publicvoid** createDataBase() **throws** IOException{

**boolean** dbExist = checkDataBase(); //kiem tra db

**if**(dbExist){

//khong lam gi ca, database da co roi

}

**else**{

**this**.getReadableDatabase();

**try** {

copyDataBase(); //chep du lieu

}

**catch** (IOException e) {

**thrownew** Error("Error copying database");

}

}

}

# Kiểm tra kết quả trước khi phát triển tiếp chương trình

Tiếp theo ta sẽ hiển thị kết quả lên textview để test xem có lấy được dữ liệu không.

1. Cũng trong file quanlycauhoi.java , xây dựng 1 hàm để đọc hết dữ liệu ra và trả về con trỏ, hàm này nhằm mục đích để test kết quả, có thể bỏ hàm này sau.

**public** Cursor laytatcacauhoi()

{

SQLiteDatabase db=**this**.getReadableDatabase();

Cursor contro=db.rawQuery("select \* from tablecauhoi", **null**);

**return** contro;

}

1. Mở file layout kéo vào 1 textView.
2. Mở file java chính (Gameshow1Activity.java) , khai báo và ánh xạ textView.
3. Trong onCreate tạo ra đối tượng quanlycauhoi. Sau đó gọi hàm createDatabase để tạo và chép dữ liệu từ database trong thư mục assets sang bộ nhớ chương trình. Do hàm createDatabase có ném lỗi nên qua đây khi gọi nó ta phải try…catch.

quanlycauhoi db=**new** quanlycauhoi(**this**);

**try** {

db.createDataBase();

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

Toast.*makeText*(**this**,"bi loi rui", Toast.*LENGTH\_SHORT*).show();

}

1. Viết tiếp để lấy dữ liệu đổ lên textView xem được hay chưa.

Cursor contro=db.laytatcacauhoi();

contro.moveToFirst();

String chuoi="";

**do**

{

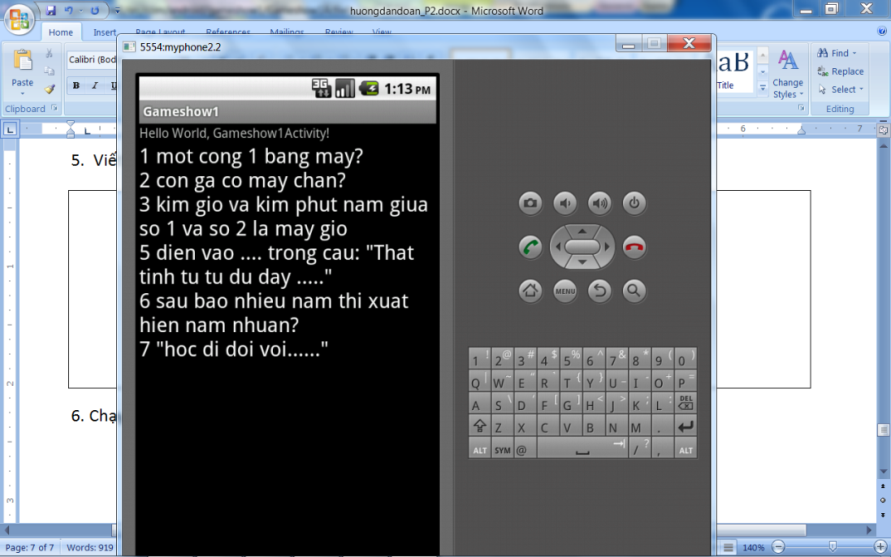
chuoi+=contro.getString(0) + " ";

chuoi+=contro.getString(1) + "\n";

}**while**(contro.moveToNext());

tv.setText(chuoi);

1. Chạy và kiểm tra kết quả như hình.



# Xây thành đối tượng và tạo hàm để lấy giới hạn và ngẫu nhiên câu hỏi

1. Tạo ra 1 class mới đặt tên “cauhoi.java” đây là lớp cho câu hỏi để dễ quản lý như sau:

**publicclass** cauhoi {

**publicint**\_id;

**public** String cauhoi, cau\_a, cau\_b, cau\_c, cau\_d, dapan ;

**public** cauhoi()

{

}

**public** cauhoi(**int** \_id,String ch,String a,String b,String c, String d, String da)

{

**this**.\_id=\_id; **this**.cauhoi=ch;

**this**.cau\_a=a; **this**.cau\_b=b;

**this**.cau\_c=c; **this**.cau\_d=d;

**this**.dapan=da;

}

}

Chú ý: để lấy dữ liệu nên xây dựng các hàm get và thuộc tính nên thành private. Tuy nhiên do làm biếng nên tôi dùng public thuộc tính cho khỏe nhưng cách này không nên.

1. Chuyển qua file “quanlycauhoi.java” viết thêm hàm để lấy ra ngẫu nhiên N câu hỏi và trả về List.

**public** List<cauhoi> layNcaungaunhien(**int** socau)

{

List<cauhoi> ds\_cauhoi=**new** ArrayList<cauhoi>();

String limit="0,"+socau;

SQLiteDatabase db=**this**.getReadableDatabase();

Cursor contro=db.query(*TABLE\_NAME*, //ten bang

**null** ,//danh sach cot can lay

**null**, //dieu kien where

**null**, //doi so dieu kien where

**null**, //bieu thuc groupby

**null**, //bieu thuc having

"random()", //bieu thuc orderby

limit//"0,3" //bieuthuc limit

);

contro.moveToFirst();

**do**{

cauhoi x=**new** cauhoi();

x.\_id=Integer.*parseInt*(contro.getString(0));

x.cauhoi=contro.getString(1);

x.cau\_a=contro.getString(2);

x.cau\_b=contro.getString(3);

x.cau\_c=contro.getString(4);

x.cau\_d=contro.getString(5);

x.dapan=contro.getString(6);

ds\_cauhoi.add(x);

}**while**(contro.moveToNext());

**return** ds\_cauhoi;

}

1. Qua file Gameshow1Activity.java. Xóa phần Cursor cũ đi và viết lại như sau:

String chuoi="";

List<cauhoi>ds\_cauhoi=**new** ArrayList<cauhoi>();

ds\_cauhoi=db.layNcaungaunhien(2);

**for**(cauhoi x: ds\_cauhoi)

{

chuoi+=x.cauhoi+"\n";

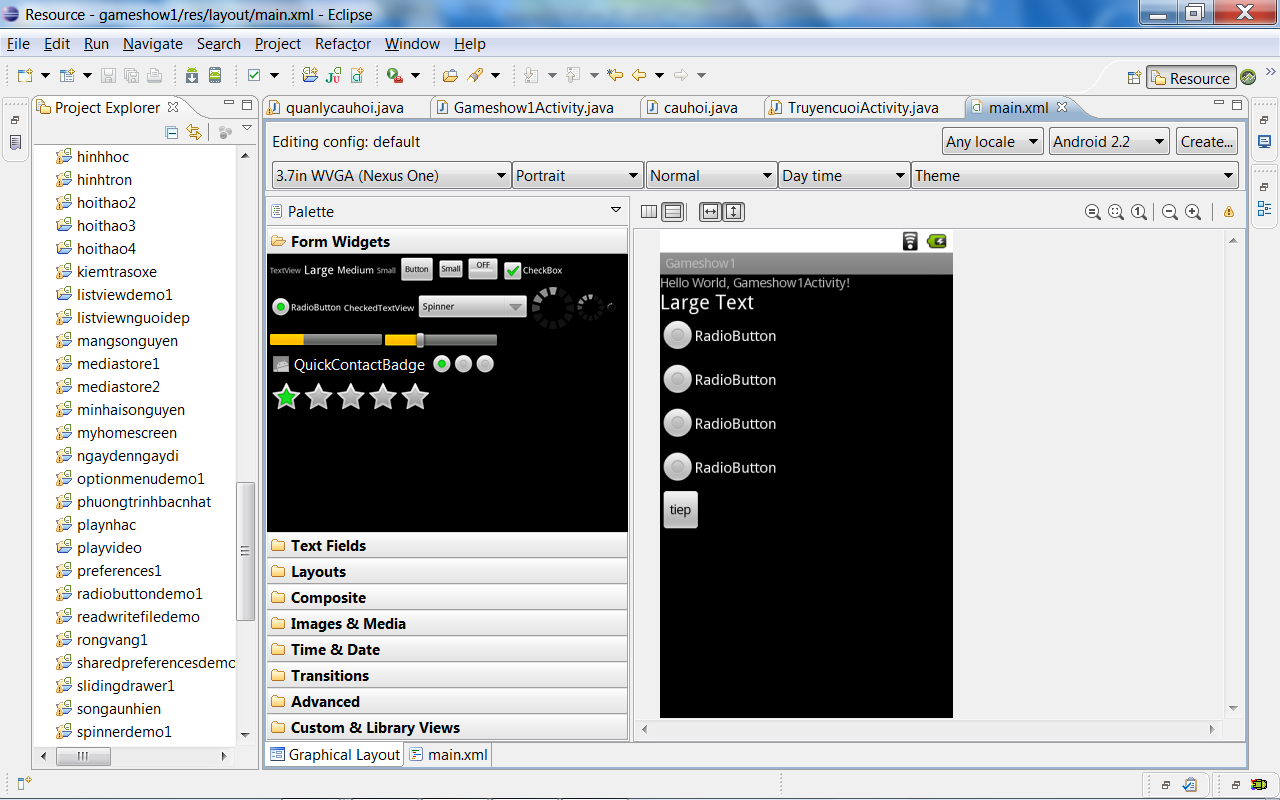
}

tv.setText(chuoi);

1. Chạy thử, thấy kết quả hiện lên được textView là OK.

# Đưa vào radio để mốc từng câu lên, trả lời và đếm câu đúng.

1. Qua giao diện kéo vào thêm 1 radiogroup có 4 radiobutton và 1 button có text “tiep”.



1. Quay lại file java chính thêm các biến và ánh xạ chúng, đồng thời bổ sung thêm các biến toàn cục như sau:

TextView tv;

**int**socau=4;

**int**index=0;

List<cauhoi>ds\_cauhoi;

RadioButton rdo\_a,rdo\_b,rdo\_c,rdo\_d;

cauhoicauhientai;

Button bt;

**int**caudung=0;

socau: là số lượng câu hỏi muốn lấy ngẫu nhiên ra từ database.

Index : dùng để lưu hiện tại đang ở câu thứ mấy.

List<cauhoi> ds\_cauhoi: dùng lưu danh sách câu được lấy ra từ database và bỏ vào đây.

Bốn radio group va button bt để nhấn câu tiếp theo (nhớ ánh xạ)

Caudung: đếm số câu đúng.

1. Xây 1 hàm mới tên “hienthi” nhận vào index và hiện thị câu hỏi trong List tại vị trí index lên.

**publicvoid** hienthi(**int** vitri)

{

cauhientai=ds\_cauhoi.get(vitri);

tv.setText(cauhientai.cauhoi);

rdo\_a.setText(cauhientai.cau\_a);

rdo\_b.setText(cauhientai.cau\_b);

rdo\_c.setText(cauhientai.cau\_c);

rdo\_d.setText(cauhientai.cau\_d);

}

1. Tiếp theo trong onCreate ta bỏ các phần hiển thị để test dữ liệu đã làm ở phần 2 (phần tạo Cursor, do..while) và thêm phần hiện thị mới như sau:

ds\_cauhoi=**new** ArrayList<cauhoi>();

ds\_cauhoi=db.layNcaungaunhien(socau);

hienthi(index);

1. Cuối cùng bắt sự kiện cho nút “tiep” để chuyển câu hỏi và xét đúng sai cũng như kết thúc:

bt.setOnClickListener(**new** View.OnClickListener() {

**publicvoid** onClick(View v) {

// **TODO** Auto-generated method stub

///xu ly phan tinh diem o day

String cautraloi="";

**if**(rdo\_a.isChecked()==**true**)

cautraloi="a";

**elseif**(rdo\_b.isChecked()==**true**)

cautraloi="b";

**elseif**(rdo\_c.isChecked()==**true**)

cautraloi="c";

**elseif**(rdo\_d.isChecked()==**true**)

cautraloi="d";

**else**

cautraloi="bo qua";

Toast.*makeText*(getApplicationContext(), "cau tra loi " + cautraloi, Toast.*LENGTH\_SHORT*).show();

**if**(cautraloi.equalsIgnoreCase(cauhientai.dapan))

{

caudung=caudung+1;

}

index++;

**if**(index<socau)

hienthi(index);

**else**

{

///xong roi ne, kich activity moi de bao ket qua

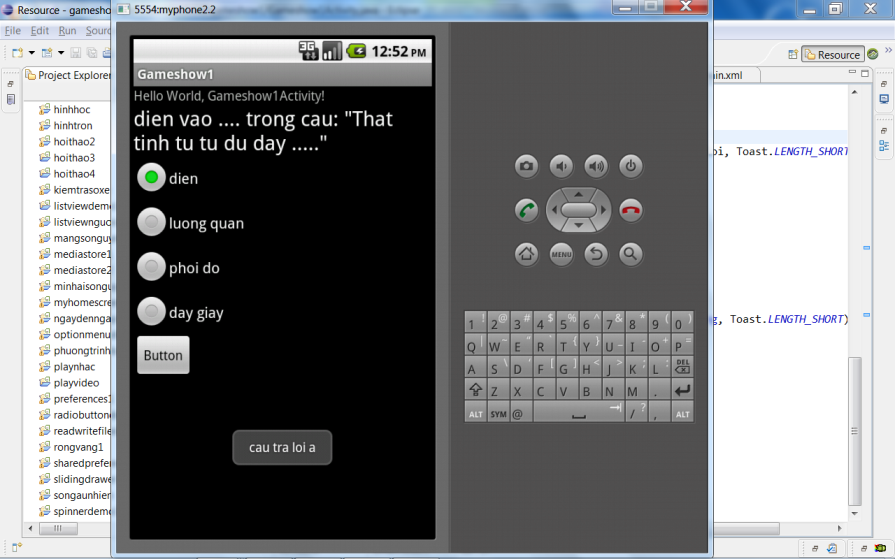
Toast.*makeText*(getApplicationContext(), "cau dung " + caudung, Toast.*LENGTH\_SHORT*).show();

}

}

});

1. Chạy chương trình để thấy kết quả.



# Viết tiếp để hoàn thiện.

Mệt rồi, đến đây tự xử nhé các bạn.