最近在处理将图片保存到sqlite[**数据库**](http://lib.csdn.net/base/mysql)中问题，在网上搜了很久，得出图片存数据库中基本以BINARY 或bolb数据类型保存 ,这两种数据类型保存都可以，相对而言blob要合适一些，因为：

BLOB (binary large object)，二进制大对象，是一个可以存储二进制文件的容器。

在计算机中，BLOB常常是数据库中用来存储[二进制文件](http://baike.baidu.com/view/1473761.htm)的字段类型。

BLOB是一个大文件，典型的BLOB是一张图片或一个声音文件，由于它们的尺寸，必须使用特殊的方式来处理（例如：上传、下载或者存放到一个数据库）。

大型对象

BLOB就是使用二进制保存数据。

如：保存位图。

\SQLite3 支持的数据类型5种：Null,Integer,Real(浮点数)，Text,BLOB(二进制对象)

BLOB 是二进制大对象（binary large object）的首字母缩写，是在 SQL Server 中作为一个单一实体存储的二进制数据集合。  
BLOB 主要用于保存多媒体对象，比如图像、视频和声音，但是它们还可以存储程序，甚至是代码片断。  
虽然 SQL Server 支持 BLOB，但不是所有数据都支持。

[**Android**](http://lib.csdn.net/base/android)数据库中存取图片通常使用两种方式，一种是保存图片所在路径，二是将图片以二进制的形式存储（sqlite3支持BLOB数据类型）。对于两种方法的使用，好像第二种方法不如第一种方法更受程序员欢迎，他们认为，在很多数据库语言里，处理大字段都是不容易的，像图片这样的文件放在数据库里会有问题：对数据库的读写速度永远赶不上文件系统的处理速度，使数据库变得巨大；但也有很多人认为像图片这样的数据存放在数据库中也有好处：易于备份，且备份速度绝对比备份文件快，比较容易数据迁移等等。其实这两种方法都有优缺点，具体使用哪种方法要视情况而定。个人倾向于使用数据库存取图片，因为个人认为存到数据库里的数据不会因外部数据的变化而丢失改变，比如你拍照获得一张图片，如果是将路径存到数据库，当这张照片被删除之后，下次读取数据库就得不到想要的结果了。接下来详细介绍数据库存取图片的方法：

1.从资源中获取Bitmap对象

1 Resources res = getResources();

2 Bitmap bmp = BitmapFactory.decodeResource(res, R.drawable.icon);

2.把图片转换成字节

public byte[] img(int id)  
{  
     ByteArrayOutputStream baos = new ByteArrayOutputStream();  
     Bitmap bitmap = ((BitmapDrawable) getResources().getDrawable(id)).getBitmap();  
     bitmap.compress(Bitmap.CompressFormat.PNG, 100, baos);  
     return baos.toByteArray();  
}

3.在数据库中插入图片

//在数据库创建时，图片字段的数据类型存储为 BLOB数据库插入操作  
public void onCreate(SQLiteDatabase db)  
{   
    String sql = "create table " + TB\_NAME + " ( " + ID + " integer primary key , " + IMAGE + " BLOB ) ";  
    db.execSQL(sql);  
}   
  
  
//将图片一字节形式存储数据库读取操作  
public long insert(byte[] img)   
{   
    SQLiteDatabase db = getWritableDatabase();  
    ContentValues cv = new ContentValues();  
    cv.put(IMAGE, img);  
    long result = db.insert(TB\_NAME, null, cv);  
    return result;  
}

4.获取存入数据库的图片（Bitmap）

public Bitmap getBmp(int position)   
{  
    SQLiteDatabase db = getReadableDatabase();  
    Cursor cursor = select(TB\_NAME);  
    cursor.moveToPosition(position);  
    byte[] in = cursor.getBlob(cursor.getColumnIndex(IMAGE));  
    Bitmap bmpout = BitmapFactory.decodeByteArray(in, 0, in.length);  
    return bmpout;  
}

//imgView.setImageBitmap(bm);

5.转换获取的图片（Bitmap）为Drawable

public Drawable chage\_to\_drawable(Bitmap bp)  
{  
    //因为BtimapDrawable是Drawable的子类，最终直接使用bd对象即可。  
    Bitmap bm=bp;   
    BitmapDrawable bd= new BitmapDrawable(getResource(), bm);   
    return bd;  
}

1、bitmap保存到SQLite 中 数据格式：

1. db.execSQL("Create table express ( \_id INTEGER PRIMARY KEY AUTOINCREMENT,express\_no varchar(100),express\_name TEXT,express\_img BLOB );");

2、bitmap 变为 Blob

1. ContentValues values = **new** ContentValues();
2. **final** ByteArrayOutputStream os = **new** ByteArrayOutputStream();
3. bmp.compress(Bitmap.CompressFormat.PNG, 100, os);
4. values.put("express\_img", os.toByteArray());
5. values.put("express\_name","zf");
6. values.put("express\_no","zf");
7. getContentResolver().insert("express", values);

 3、从SQLite中读取Bitmap

1. **byte**[] in=cur.getBlob(cur.getColumnIndex("express\_img"));
2. bmpout=BitmapFactory.decodeByteArray(in,0,in.length);

 显示在ImageView上

ImageView imageView = (ImageView) view.findViewById(R.id.img);

1. ByteArrayInputStream stream = **new** ByteArrayInputStream(cur.getBlob(cur.getColumnIndex("express\_img")));
2. imageView.setImageDrawable(Drawable.createFromStream(stream, "img"));

总结：

inputStream:  作为数据缓存，数据写如何供别的对象读取，其方法为read();

outputStream:作为数据缓存，将来向别的对象写内容!其方法write();

1. **byte**[] in=cur.getBlob(cur.getColumnIndex(MyUser.User.BITMAP\_VALUES));

 //这样也可以对数据进行初始化，byte是基本类型，不需要之前进行长度定义。

一般只保存图片的路径，没有直接把图片保存进去，我以前也有试过想放图片到SQlite数据库，后来图片才放几张就不能放了。

sqlite数据库到达一定大的情况下就不能往里面加数据

比如说你把image.jpg放在sdcard的images目录下，那你就在数据库存/sdcard/images/image.jpg

（ 我是直接存图片名字，然后调用的时候再补上图片的路径！代码如下private String MUSIC\_PATH = new String("/sdcard/feiyang/voice/");

然后根据需要写上所要的图片名称。

使用SQLite数据库中的blob类型来存储，可以参考下面两个网址：

如何使用blob类型存储mp3文件：<http://doc.chinaunix.net/android/200903/164048.shtml>

android操作sqlite3的blob字段：<http://marshal.easymorse.com/archives/2342>

1、bitmap保存到SQLite 中 数据格式：

     db.execSQL("Create table " + TABLE\_NAME + "( \_id INTEGER PRIMARY KEY AUTOINCREMENT,USER\_AGE  INTEGER,USER\_NAME TEXT,BITMAP\_VALUES BLOB );");

2、bitmap 变为 Blob

    ContentValues values = new ContentValues();

    final ByteArrayOutputStream os = new ByteArrayOutputStream();

    bmp.compress(Bitmap.CompressFormat.PNG, 100, os);

    values.put(MyUser.User.BITMAP\_VALUES, os.toByteArray());

    values.put(MyUser.User.USER\_NAME,"icon");

    values.put(MyUser.User.USER\_AGE,50);

    getContentResolver().insert(MyUser.User.CONTENT\_URI, values);

3、从SQLite中读取Bitmap

     byte[] in=cur.getBlob(cur.getColumnIndex(MyUser.User.BITMAP\_VALUES));

     bmpout=BitmapFactory.decodeByteArray(in,0,in.length);

总结：

inputStream:  作为数据缓存，数据写如何供别的对象读取，其方法为read();

outputStream:作为数据缓存，将来向别的对象写内容!其方法write();

byte[] in=cur.getBlob(cur.getColumnIndex(MyUser.User.BITMAP\_VALUES));//这样也可以对数据进行初始化，byte是基本类型，不需要之前进行长度定义

存储图片：bitmap

1. **private** **byte**[] getIconData(Bitmap bitmap){
3. **int** size = bitmap.getWidth()\*bitmap.getHeight()\*4;
5. ByteArrayOutputStream out = **new** ByteArrayOutputStream(size);
7. **try** {
9. bitmap.compress(Bitmap.CompressFormat.PNG, 100, out);
11. out.close();
13. } **catch** (IOException e) {
15. e.printStackTrace();
17. }
19. **return** out.toByteArray();
21. }

获取图片：

1. Bitmap getIconFromCursor(Cursor c, **int** iconIndex) {
3. **byte**[] data = c.getBlob(iconIndex);
5. **try** {
7. **return** BitmapFactory.decodeByteArray(data, 0, data.length);
9. } **catch** (Exception e) {
11. **return** **null**;
13. }
15. }