<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
   xmlns:tools= "http://schemas.android.com/tools"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent"  
   android:orientation="vertical"  
   tools:context="com.example.saveimage.MainActivity" >  
   <EditText  
 android:id="@+id/txtedit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
   
 />  
     <Button  
 android:id="@+id/btdownload"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Download and save to database"   
 android:onClick="downloadImage"  
 />  
   
 <TextView  
 android:id="@+id/txtview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:singleLine="false"  
 android:layout\_gravity="center"  
   
 />  
 <ImageView  
 android:id="@+id/imgview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 />  
  
</LinearLayout>

Then in the src folder, your create a class called DataHelper that helps you in creating a sqlite database and a table to store images, inserting image data to the database, and reading the image data to display in the ImageView. 

import java.io.ByteArrayOutputStream;  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteException;  
import android.database.sqlite.SQLiteOpenHelper;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.util.Log;  
  
public class DataHelper extends SQLiteOpenHelper {  
 public static final String DATABASE\_NAME = "imgdb";  
 public static final String TABLE\_NAME = "tbl\_img";  
 public static final int DATABASE\_VERSION = 1;  
 public static final String CREATE\_TABLE = "CREATE TABLE IF NOT EXISTS "+ TABLE\_NAME+ "(id INTEGER PRIMARY KEY AUTOINCREMENT, img BLOB NOT NULL, description TEXT NULL)";   
 public static final String DELETE\_TABLE="DROP TABLE IF EXISTS " + TABLE\_NAME;  
  
public DataHelper(Context context) {  
 super(context, DATABASE\_NAME, null, DATABASE\_VERSION);  
   
}   
public void onCreate(SQLiteDatabase db) {  
 // Create the table  
 db.execSQL(CREATE\_TABLE);  
   
 }  
//Upgrading database   
@Override   
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 //Drop older table if existed  
 db.execSQL(DELETE\_TABLE);  
 //Create tables again  
 onCreate(db);  
 }  
  
public void insertBitmap(Bitmap bm) {  
   
 // Convert the image into byte array  
 ByteArrayOutputStream out = new ByteArrayOutputStream();  
 bm.compress(Bitmap.CompressFormat.PNG, 100, out);  
 byte[] buffer=out.toByteArray();  
 // Open the database for writing  
 SQLiteDatabase db = this.getWritableDatabase();  
 // Start the transaction.  
 db.beginTransaction();  
 ContentValues values;  
  
 try  
 {  
 values = new ContentValues();  
 values.put("img", buffer);  
 values.put("description", "Image description");  
 // Insert Row  
 long i = db.insert(TABLE\_NAME, null, values);  
 Log.i("Insert", i + "");  
 // Insert into database successfully.  
 db.setTransactionSuccessful();  
  
 }  
 catch (SQLiteException e)  
 {  
 e.printStackTrace();  
  
 }  
 finally  
 {  
 db.endTransaction();  
 // End the transaction.  
 db.close();  
 // Close database  
 }   
 }  
  
 public Bitmap getBitmap(int id){  
 Bitmap bitmap = null;  
 // Open the database for reading  
 SQLiteDatabase db = this.getReadableDatabase();  
 // Start the transaction.  
 db.beginTransaction();  
  
 try  
 {  
 String selectQuery = "SELECT \* FROM "+ TABLE\_NAME+" WHERE id = " + id;  
 Cursor cursor = db.rawQuery(selectQuery, null);  
 if(cursor.getCount() >0)  
 {  
 while (cursor.moveToNext()) {  
 // Convert blob data to byte array  
 byte[] blob = cursor.getBlob(cursor.getColumnIndex("img"));  
 // Convert the byte array to Bitmap  
 bitmap=BitmapFactory.decodeByteArray(blob, 0, blob.length);  
  
 }  
  
 }  
 db.setTransactionSuccessful();  
  
 }  
 catch (SQLiteException e)  
 {  
 e.printStackTrace();  
  
 }  
 finally  
 {  
 db.endTransaction();  
 // End the transaction.  
 db.close();  
 // Close database  
 }  
 return bitmap;  
  
}   
  
}  
  
In the MainActivity class, you write code to download an image from the internet, save it in the database, and read the image to from the database to display in the ImageView.

package com.example.inputdialog;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import android.app.Activity;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.TextView;

public class MainActivity extends Activity

{  
 private DataHelper dbHelper;  
 private EditText et;  
   protected void onCreate(Bundle savedInstanceState) {  
     super.onCreate(savedInstanceState);  
     setContentView(R.layout.activity\_main);  
     et=(EditText)findViewById(R.id.txtedit);  
     dbHelper=new DataHelper(this);  
           
   }  
     
   public void downloadImage(View view)

{  
   BackTask bt=new BackTask();  
   String imgURL=et.getText().toString();  
   if(!imgURL.trim().equals(""))

{  
   bt.execute(imgURL);  
   }  
   }  
     
 // AsynnTask to run download an image in background

   private class BackTask extends AsyncTask<String,Void,Bitmap>{   
   TextView tv;  
   protected void onPreExecute(){  
     tv=(TextView)findViewById(R.id.txtview);  
     tv.setText("Downloading the image. Please wait...");  
   }  
   protected Bitmap doInBackground(String...params){  
   Bitmap bitmap=null;  
   try {  
   // Download the image  
     URL url = new URL(params[0]);  
     HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
     connection.setDoInput(true);  
     connection.connect();  
     InputStream is = connection.getInputStream();  
     // Decode image to get smaller image to save memory   
     final BitmapFactory.Options options = new BitmapFactory.Options();   
     options.inJustDecodeBounds = false;  
     options.inSampleSize=4;  
     bitmap = BitmapFactory.decodeStream(is,null, options);  
     is.close();  
     }  
     catch(IOException e){  
     return null;  
     }  
   return bitmap;  
   }  
   protected void onPostExecute(Bitmap result){  
   tv.setVisibility(TextView.GONE);  
   // Insert bitmap to the database  
   dbHelper.insertBitmap(result);  
   ImageView imgView=(ImageView)findViewById(R.id.imgview);   
   // Read the first image from database and show it in ImageView     
   imgView.setImageBitmap(dbHelper.getBitmap(1));  
     }  
   }  
     
}