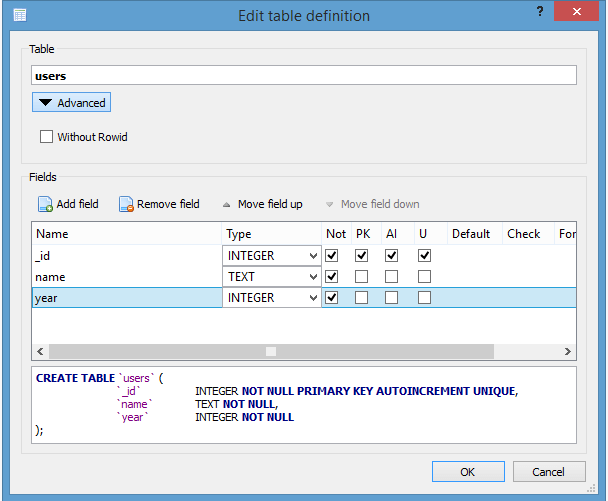
**Использование существующей БД SQLite**

Кроме создания новой базы данных мы также можем использовать уже существующую. Это может быть более предпочтительно, так как в этом случае база данных приложения уже будет содержать всю необходимую информацию.

Возьмем проект, созданный в предыдущих темах. Для начала создадим базу данных SQLite. В этом нам может помочь такой инструмент как **Sqlitebrowser**. Он бесплатный и доступен для различных операционных систем по адресу <http://sqlitebrowser.org/>. Хотя можно использовать и другие способы для создания начальной БД.

Sqlitebrowser представляет графический интерфейс для создания базы данных и определения в ней всех необходимых таблиц:



Как видно на скриншоте, я определяю таблицу users с тремя полями: \_id, name, age. Общая команда на создание таблицы будет следующей:

CREATE TABLE `users` (

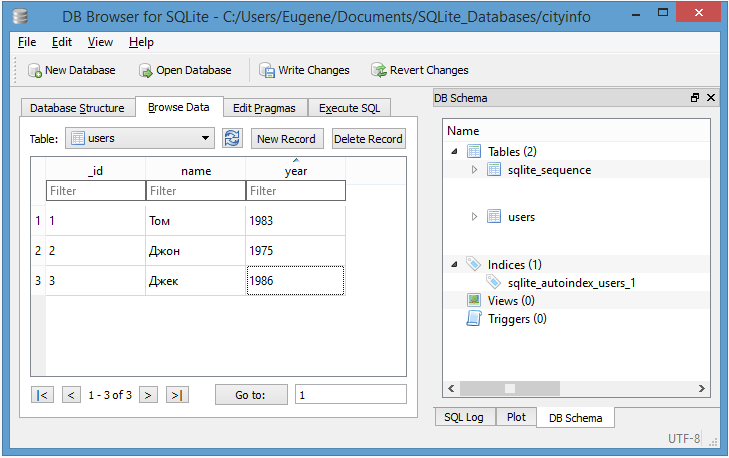
`\_id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,

`name` TEXT NOT NULL,

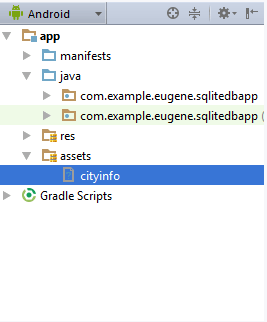
`year` INTEGER NOT NULL

);

Там же в программе добавим несколько элементов в созданную таблицу:



После создания таблицы добавим в проект в Android Studio папку **assets**, а в папку assets - только что созданную базу данных:



В моем случае база данных называется "cityinfo".

Изменим код DatabaseHelper следующим образом:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84 | package com.example.eugene.sqlitedbapp;    import android.database.sqlite.SQLiteOpenHelper;  import android.database.sqlite.SQLiteDatabase;  import android.content.Context;  import java.io.IOException;  import java.io.InputStream;  import java.io.OutputStream;  import java.io.FileOutputStream;  import java.io.File;  import java.sql.SQLException;    public class DatabaseHelper extends SQLiteOpenHelper {        private static String DB\_PATH = "/data/data/com.example.eugene.sqlitedbapp/databases/";      private static String DB\_NAME = "cityinfo";      private static final int SCHEMA = 1; // версия базы данных      static final String TABLE = "users";        public static final String COLUMN\_ID = "\_id";      public static final String COLUMN\_NAME = "name";      public static final String COLUMN\_YEAR = "year";      public SQLiteDatabase database;      private Context myContext;        public DatabaseHelper(Context context) {          super(context, DB\_NAME, null, SCHEMA);          this.myContext=context;      }        @Override      public void onCreate(SQLiteDatabase db) {      }      @Override      public void onUpgrade(SQLiteDatabase db, int oldVersion,  int newVersion) {        }        public void create\_db(){          InputStream myInput = null;          OutputStream myOutput = null;          try {              File file = new File(DB\_PATH + DB\_NAME);              if (!file.exists()) {                  this.getReadableDatabase();                  //получаем локальную бд как поток                  myInput = myContext.getAssets().open(DB\_NAME);                  // Путь к новой бд                  String outFileName = DB\_PATH + DB\_NAME;                    // Открываем пустую бд                  myOutput = new FileOutputStream(outFileName);                    // побайтово копируем данные                  byte[] buffer = new byte[1024];                  int length;                  while ((length = myInput.read(buffer)) > 0) {                      myOutput.write(buffer, 0, length);                  }                    myOutput.flush();                  myOutput.close();                  myInput.close();              }          }          catch(IOException ex){            }      }      public void open() throws SQLException {          String path = DB\_PATH + DB\_NAME;          database = SQLiteDatabase.openDatabase(path, null,                      SQLiteDatabase.OPEN\_READWRITE);        }        @Override      public synchronized void close() {          if (database != null) {              database.close();          }          super.close();      }  } |

По умолчанию база данных будет размещаться во внешнем хранилище, выделяемом для приложения в папке */data/data/[название\_пакета]/databases/*, поэтому мы и используем данный путь с указанием пакета приложения.

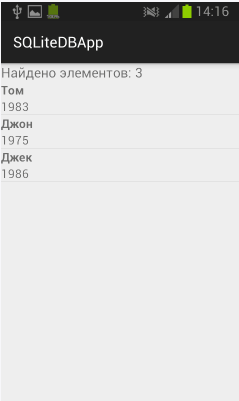
Метод onCreate() нам не нужен, так как нам не требуется создание встроенной базы данных. Зато здесь определен дополнительный метод create\_db(), цель которого копирование базы данных из папки assets в то место, которое указано в переменной DB\_PATH.

Кроме этого здесь также определен метод открытия базы данных open() с помощью методаSQLiteDatabase.openDatabase()

Новый способ организации подключения изменит использование DatabaseHelper в activity. Так, чтобы получить данные в MainActivity, теперь нам надо обновить ее код:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87 | package com.example.eugene.sqlitedbapp;    import android.content.Intent;  import android.support.v7.app.ActionBarActivity;  import android.os.Bundle;  import android.view.Menu;  import android.view.MenuItem;    import android.view.View;  import android.widget.AdapterView;  import android.widget.TextView;  import android.widget.ListView;  import android.database.Cursor;  import android.widget.SimpleCursorAdapter;  import java.sql.SQLException;    public class MainActivity extends ActionBarActivity {        ListView mList;      TextView header;      DatabaseHelper sqlHelper;      Cursor userCursor;      SimpleCursorAdapter userAdapter;        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_main);            header = (TextView)findViewById(R.id.header);          mList = (ListView)findViewById(R.id.list);          mList.setOnItemClickListener(new AdapterView.OnItemClickListener() {              @Override              public void onItemClick(AdapterView<?> parent, View view, int position, long id) {                    Intent intent = new Intent(getApplicationContext(), UserActivity.class);                  intent.putExtra("id", id);                  startActivity(intent);              }          });          sqlHelper = new DatabaseHelper(getApplicationContext());          // создаем базу данных          sqlHelper.create\_db();      }      @Override      public void onResume(){          super.onResume();          try {              sqlHelper.open();              userCursor = sqlHelper.database.rawQuery("select \* from " + DatabaseHelper.TABLE, null);              String[] headers = new String[]{DatabaseHelper.COLUMN\_NAME, DatabaseHelper.COLUMN\_YEAR};              userAdapter = new SimpleCursorAdapter(this, android.R.layout.two\_line\_list\_item,                      userCursor, headers, new int[]{android.R.id.text1, android.R.id.text2}, 0);              header.setText("Найдено элементов: " + String.valueOf(userCursor.getCount()));              mList.setAdapter(userAdapter);          }          catch (SQLException ex){}      }        @Override      public void onDestroy(){          super.onDestroy();          // Закрываем подключения           sqlHelper.database.close();           userCursor.close();      }        @Override      public boolean onCreateOptionsMenu(Menu menu) {            getMenuInflater().inflate(R.menu.menu\_main, menu);          return true;      }        @Override      public boolean onOptionsItemSelected(MenuItem item) {            int id = item.getItemId();          if (id == R.id.action\_settings) {                Intent intent = new Intent(getApplicationContext(), UserActivity.class);              startActivity(intent);              return true;          }          return super.onOptionsItemSelected(item);      }  } |

Все взаимодействие с базой данных теперь будет идти через объект sqlHelper.database, представляющий SQLiteDatabase, но теперь заключенный в самом классе DatabaseHelper.



И также изменится UserActivity, которая предоставляет функционал для добавления, обновления и удаления данных:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138 | package com.example.eugene.sqlitedbapp;    import android.content.ContentValues;  import android.database.Cursor;  import android.support.v7.app.ActionBarActivity;  import android.support.v4.app.Fragment;  import android.os.Bundle;  import android.view.LayoutInflater;  import android.view.Menu;  import android.view.MenuItem;  import android.view.View;  import android.view.ViewGroup;  import android.widget.Button;  import android.widget.EditText;  import android.content.Intent;  import java.sql.SQLException;    public class UserActivity extends ActionBarActivity {        @Override      protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.activity\_user);            long userId=0;          Bundle extras = getIntent().getExtras();          if (extras != null) {                userId = extras.getLong("id");          }            if (savedInstanceState == null) {              getSupportFragmentManager().beginTransaction()                      .add(R.id.container, PlaceholderFragment.newInstance(userId))                      .commit();          }      }        public static class PlaceholderFragment extends Fragment {            EditText nameBox;          EditText yearBox;          Button delButton;          Button saveButton;            DatabaseHelper sqlHelper;          Cursor userCursor;            public static PlaceholderFragment newInstance(long id) {              PlaceholderFragment fragment = new PlaceholderFragment();              Bundle args=new Bundle();              args.putLong("id", id);              fragment.setArguments(args);              return fragment;          }            @Override          public void onCreate(Bundle savedInstanceState) {              super.onCreate(savedInstanceState);              setRetainInstance(true);              }          public PlaceholderFragment() {          }            @Override          public View onCreateView(LayoutInflater inflater, ViewGroup container,                                   Bundle savedInstanceState) {              View rootView = inflater.inflate(R.layout.fragment\_user, container, false);              nameBox = (EditText) rootView.findViewById(R.id.name);              yearBox = (EditText) rootView.findViewById(R.id.year);              delButton = (Button) rootView.findViewById(R.id.delete);              saveButton = (Button) rootView.findViewById(R.id.save);                final long id = getArguments() != null ? getArguments().getLong("id") : 0;              sqlHelper = new DatabaseHelper(getActivity());              try {                    sqlHelper.open();                    // кнопка удаления                  delButton.setOnClickListener(new View.OnClickListener() {                      @Override                      public void onClick(View v) {                            sqlHelper.database.delete(DatabaseHelper.TABLE, "\_id = ?",                                                      new String[]{String.valueOf(id)});                          goHome();                      }                  });                    // кнопка сохранения                  saveButton.setOnClickListener(new View.OnClickListener() {                      @Override                      public void onClick(View v) {                          ContentValues cv = new ContentValues();                          cv.put(DatabaseHelper.COLUMN\_NAME, nameBox.getText().toString());                          cv.put(DatabaseHelper.COLUMN\_YEAR, Integer.parseInt(yearBox.getText().toString()));                            if (id > 0) {                              sqlHelper.database.update(DatabaseHelper.TABLE, cv,                                      DatabaseHelper.COLUMN\_ID + "=" + String.valueOf(id), null);                          } else {                              sqlHelper.database.insert(DatabaseHelper.TABLE, null, cv);                          }                          goHome();                      }                  });                    // если 0, то добавление                  if (id > 0) {                      // получаем элемент по id из бд                      userCursor = sqlHelper.database.rawQuery("select \* from " + DatabaseHelper.TABLE + " where " +                              DatabaseHelper.COLUMN\_ID + "=?", new String[]{String.valueOf(id)});                      userCursor.moveToFirst();                      nameBox.setText(userCursor.getString(1));                      yearBox.setText(String.valueOf(userCursor.getInt(2)));                      userCursor.close();                  } else {                      // скрываем кнопку удаления                      delButton.setVisibility(View.GONE);                  }              }              catch (SQLException ex) {                  ex.printStackTrace();              }              return rootView;          }          public void goHome(){                sqlHelper.database.close();              Intent intent = new Intent(getActivity(), MainActivity.class);              intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP | Intent.FLAG\_ACTIVITY\_SINGLE\_TOP);              startActivity(intent);          }      }  } |