Android ORM数据库的使用

`Android ORM` `ORM` `ActiveAndrod`

ORM数据库框架ActiveAndrod的特点和优势

什么是ORM框架

ORM(Object Relational Mapping)框架采用元数据来描述对象与关系映射细节。
就是利用Java的反射机制把对象和数据库记录映射关联起来。

ActiveAndroid的特点和优势

- **基于ORM关系操作数据库**
- **配置方便**
- **几乎不需要编写任何SQL语句就能够保存和检索SQLite数据库记录**
- **每一个操作都封装为一个类，save()和delete()**
- **对象形式存取数据**

ActiveAndroid的基本用法

基本配置

- **配置**
  AndroidMainfast.xml中配置

```
<application
        android:name=".MyApplication"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <!--定义数据库的name和version-->
        <meta-data android:name="AA_DB_NAME"
android:value="Pickrand.db"/>
        <meta-data android:name="AA_DB_VERSION"
android:value="5"/>

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN"
/>

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

Application.java中配置

```
public class MyApplication extends Application{
    @Override
    public void onCreate(){
        super.onCreate();
        ActiveAndroid.initialize(this);

    }
}
```

- **定义实体类**
  实体类需要继承Model,可自定义表名和属性对应的字段名
  定义实体类Category

```
@Table(name = "Categories")
public class Category extends Model {
    @Column(name = "Name")
    public String name;
}
```

定义实体类Item

```
//定义表名
    @Table(name="Items")
public class Item extends Model {
    //定义表字段
    @Column(name="Name")
    public String name;

    @Column(name="Category")
    public Category category;

    public Item(){
        super();
    }
    public Item(String name,Category category){
        super();
        this.name=name;
        this.category=category;
    }

}
```

- **进行增删改查操作**
  增加数据

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //save()插入、修改（更新）数据
        Category category=new Category();
        category.name="OutLook";
        category.save();//将数据保存
        Item item=new Item();
        item.name="QDH";
        item.category=category;
        item.save();//将数据保存
    }
```

查询数据

```
//查询数据 execute()查询出的为一集合，executeSingle()一条数据
        String select = null;
        List list =new
Select().from(Item.class).where("Name=?","QDH").orderBy("Name
ASC").execute();
        for (int i = 0; i <list.size() ; i++) {
            select+=list.get(i);
        }
        Log.i("info",select);
```

删除数据(三种方法)

```
//删除数据(删除id为1的数据)
  Item item1 =Item.load(Item.class,1);
  item1.delete();

//Item.delete(Item.class,1);//删除方法二
//new Delete().from(Item.class).where("Name=?","QDH").execute();//
删除方法三：删除特定条件的数据
```

升级数据库

- setp1

```
<!--修改升级数据库的版本号-->
 <meta-data android:name="AA_DB_VERSION"
 android:value="6" />
```

- step2
  在main/assets下新建文件夹migrations，并在其内添加sql文件，文件名为版本号的value值.sql

- step3
  编写升级SQl

```
/*在category中添加一个字段StuId*/
ALTER TABLE Categories ADD COLUMN StuId INTEGER;
```

- step4
  同时修改对应的实体（表）

```
@Table(name = "Categories")
public class Category extends Model {
    @Column(name = "Name")
    public String name;

    //升级数据库添加sql文件后同时也要在此添加字段
    @Column(name = "StuId")
    public int stuId;
}
```

ActiveAndroid的实际应用案例