GreenDao

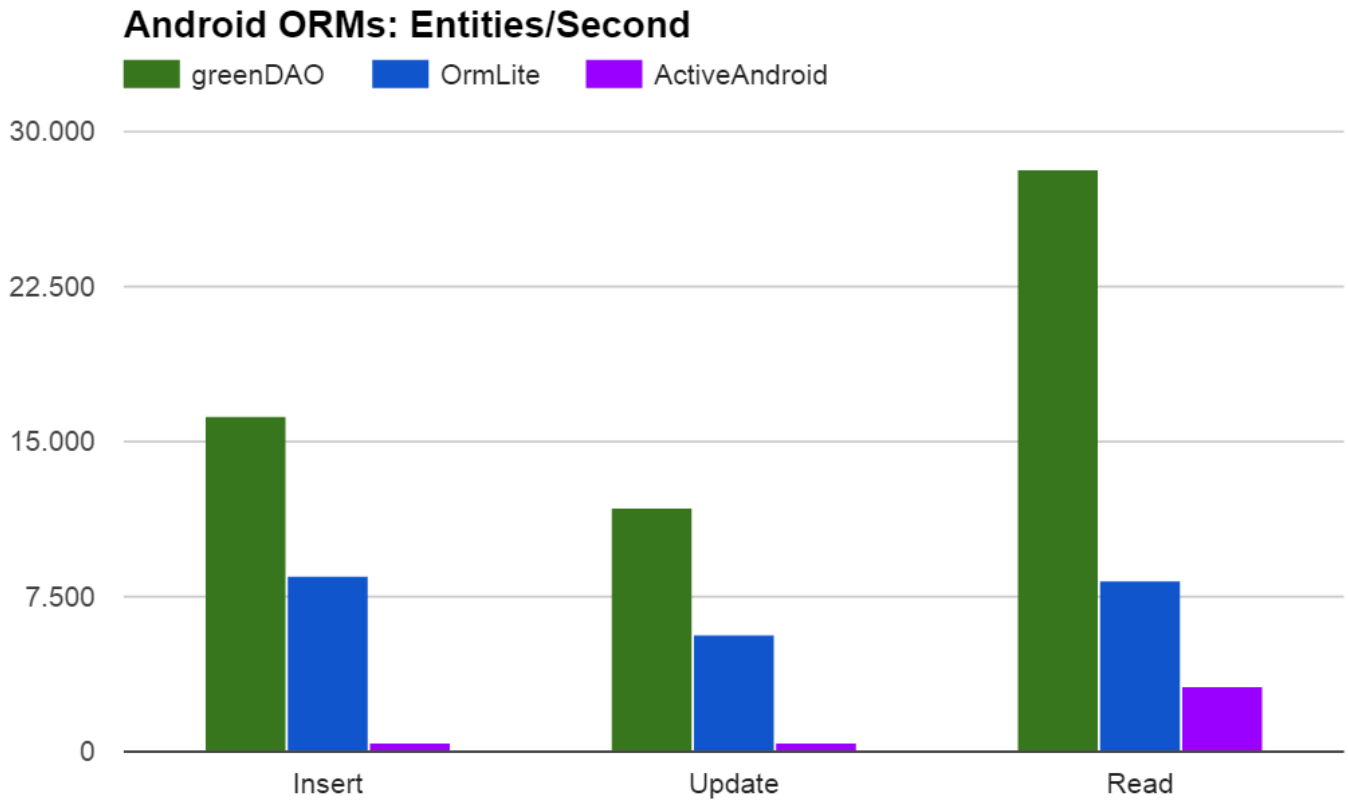
介绍

greenDao是一个将对象映射到SQLite数据库中的轻量且快速的开源的ORM解决方案。

特点

1.最大的性能，可能是android中最快ORM Database解决方案
2.易使用，只需要定义data model，GreenDao会构建data objects(Entities)和DAOS(data access objects)
3.最少的内存开销
4.依赖的库很小，< 100kb
5.支持数据加密
6.强大的社区

官方给出的性能对比



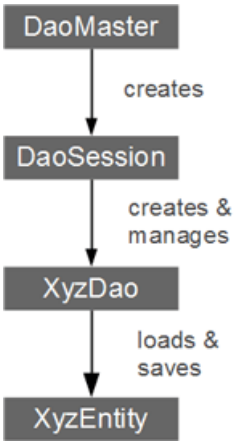
GreenDao配置

```
// In your root build.gradle file:
buildscript {
    repositories {
        jcenter()
        mavenCentral() // add repository
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.3.0'
        classpath 'org.greenrobot:greendao-gradle-plugin:3.2.2' // add plugin
    }
}

// In your app projects build.gradle file:
apply plugin: 'com.android.application'
apply plugin: 'org.greenrobot.greendao' // apply plugin

dependencies {
    compile 'org.greenrobot:greendao:3.2.2' // add library
}
```

核心类



1.**DaoMaster**

这个类是使用greenDao的入口，DaoMaster持有着SQLite数据库对象并管理着所有一个schema中的所有Dao类。它有一些静态方法用来创建表和删除表，它的内部类OpenHelper和DevOpenHelper继承自SQLiteOpenHlper用来在SQLite中创建schema

2.**DaoSession**

管理了一个schema中的所有Dao对象，可以通过getter方法获取这些Dao对象。DaoSession提供entity的一些持久化方法如insert，load，update，refresh，delete，这些方法都是由插件自动生成的

3.**DAOS**

Data access objects简称Dao，用来对entity的存取访问。针对每一个entity，greendao都会生成一个对应的DAO类，这些类提供了比DaoSession更多的持久化方法，如count，loadAll等

4.**Entity**

持久化类，通常情况下一个持久化类对应着数据库中的一张表，每条数据对应着普通的pojo类或者

javabean

> 注意了，DaoMaster,Daosession,DAOS类是默认没有的，我们需要先编写一个Entity，而Entiry的内容也只是需要编写成员变量，然后Build->makeProject，上面四个核心类就会自动构建出来，Entity的内容也会自动完善好。

GreenDao使用

官网引导

ToOne ToMany的关系介绍

```
@Entity
public class Order {
    @Id private Long id;

    private long customerId;

    @ToOne(joinProperty = "customerId")
    private Customer customer;
}

@Entity
public class Customer {
    @Id private Long id;
}
```

```
@Entity
public class Customer {
    @Id private Long id;

    @ToMany(referencedJoinProperty = "customerId")
    @OrderBy("date ASC")
    private List<Order> orders;
}

@Entity
public class Order {
    @Id private Long id;
    private Date date;
    private long customerId;
}
```

1.此时使用OrderDao的queryBuilder进行查询，默认是不会蒋Customer的数据加载进来的，只有在使用了getCustomer方法时，才会将数据加载进来，这样当不需要Customer数据时，可以很快的进行加载。

2.但是如果我需要在查询的时候就将Customer的数据加载进来，那么就要使用queryDeep方法了。queryDeep的第一个参数就是where的内容，第二个参数就是第一个参数中使用了？表示的数据，通配符数据，可以使用数组。在这里需要注意，where中使用到的 "表名.字段名" ,表名不能使用OrderDao.TABLENAME,因此在底层的查询数据中，已经将这个表名换了一个别名,在本表中，表名换成了T,ToOne,ToMany关联的表，根据顺序定义别名T0,T1,T2，以此类推。

loadDeep举例

```
List<Student> students = mStu.queryDeep("where T." + StudentDao.Properties.Name.colu
+ " and T." + StudentDao.Properties.Age.columnName + " > ?", new String[]{stuName, a
```

实际性能测试

测试的表内容
Student表 （学生）

```
    @Id(autoincrement = true)
    private Long id;

    private String name;

    private Integer age;

    private long subjectid;

    @ToOne(joinProperty = "subjectid")
    private Subject subject;
```

Subject表（课程）

```
    @Id(autoincrement = true)
    private Long id;

    private String name;

    private String teacher;
```

操作	耗时/单位：毫秒
批量插入Subject 10000	269
批量插入Student 10000,每个student对应一个Subject	224
批量插入Student 10000，共用一个Subject	253
逐个插入Subject，10000条	31568
在表Subject中60006条单个条件查找	20
在表Subject中60006条2个条件查找	14
在表Student中40009条单条件查找	32
在表Student中40009条2个条件查找	8
在表Subject中60009条单条件删除	47
在表Subject中60008条2条件删除	29

操作	耗时/单位：毫秒
操作	**耗时/单位：毫秒**
在表Student中40009条1条件删除	23
在表Student中40007条2条件删除	40
在表Student中41005条2条件删除	25
在表Subject中60310条批量更新300条	104
在表Student中41505条批量更新500	85

总结

经过测试，GreenDao的性能确实很好。