

前言

众所周知后端工程师又称为CURD工程师，所以这期任务我们就来了解下CURD。由于出题人SQL语句极其不熟练，所以为了防止检查新人任务时出现尴尬画面，所以我们就**不直接用**SQL来做CURD了（😂😂）。不废话了，引出这周的主题---- ORM。

什么是ORM

- 我也说不清（😂😂），不如贴大佬的链接吧～～：[阮一峰](#)

任务要求

- 自己实现 一个的ORM框架
- 具体要求
 - Create
 - 当自定义的Object在数据库不存在对应的表时，支持在数据库中创建表。Object 与数据库表的字段对应关系如下（用下划线分割单词）：



- 支持Object实例的创建
- 表的创建支持索引，主键
- Update
 - 对存在数据进行整体更新，不存在数据做一个错误处理
- Retrieve
 - 默认通过Object的主键子段查找
 - 指定字段查询，将数据映射到Object，支持批量查询（返回结果应该为 对象数组）
- Delete
 - 默认通过Object的主键字段查找删除
 - 支持批量删除

进一步的解释说明

- 由于前面的要求可能会造成歧义，这里用代码的形式来描述一下，在这之前说一下实现ORM的思路
 - 用反射实现ORM，Java的ORM框架Mybatis，Hibernate，Go中Gorm，即通过反射获取Object的对象信息，然后转化成SQL语句
- 实例

```
package model

//MyObject 是自定义的Object, `json:"id" myorm:"column: id;"` 是对于该字段
的一些tag
//可以把Object在数据库表中对应的信息放在里面，类似于Java的Annotation
//Python的装饰器Decorator(python不是很熟悉，如果说出了请指正)
//信息表示格式可以自行定义
//支持定义的字段属性：主键(必须指定)、索引、数据类型、是否非空(支持设定默认值)、
是否不可重复、
//未显示指定类型时 默认转化为：string -> varchar int-> int ,int64-
>bigint
type MyObject struct {
    ID          int    `myorm:"column:id;primary_id"`
    Price       int    `myorm:"column:int"`
    ObjectName  string
    `myorm:"column:object_name;type:varchar(50);index"`
    Info        string `myorm:"column:info";type:text;`
}

/*
定义了以上Object后，希望你实现以下功能
模式一，实现了一个DB类，这个类具有以下能力
*/
myObject := &MyObject{ID: 1, ObjectName: "object_name", Info:
"info"}

//1.通过传入一个Object实例，创建一个表，如果表存在 则不创建，但是需要让调用者知
道(抛出错误或者返回值)
DB.CreateTable(myObject)
//2.表中创建一个实例记录
DB.Create(myObject)
//3.更新表数据
DB.Update(myObject)
//4.查询表数据 Find 的函数声明应该和这个类似：
```

```

// func Find(whereExpression string,queryParams
...interface{}),dumpInstance interface{})(bool,error);
DB.FindOne("id = ?",myObject.ID,&myObject);
// 或者 func Find(whereExpression string,queryParams ...interface{})
(interface{}),error);
//5.批量查询
// func FindAll(whereExpression string,queryParams
[]interface{}),dumpInstance interface{} error;
// func FindAll(whereExpression string,queryParams ...interface{})
([]interface{}),error);
res := make([]*MyObject, 0)
DB.FindAll("price = ?",15,res)
//6.Delete
// func Delete(objToDel interface{}) error;
DB.Delete(myObject)
//7.DeleteAll
// func DeleteAll(whereExpression string,queryParams
...interface{} error;
Db.DeleteAll("id in ()",[]int64{1,2,3});
// 进阶
// 看一下5 6的查询，其实分析一下我们可以把接口设计的更加优雅
// func FindAll(whereExpression string,queryParams
[]interface{}),dumpInstance interface{} error; 这样的接口实在太丑了
//我们可以改造一下把它做成链式调用 如
// func where(whereExpression string,queryParams ...interface{}) DB;
// func FindOne(dumpInstance interface{} error ;
DB.where().FindOne()
// FindAll 同理
Db.where().FindAll()

```

要求

- 语言不限
- 一星期
- 不许直接粘贴网上的代码