Atitit golan conn mysql

go get -u github.com/go-sql-driver/mysql

# **[Go连接MYSQL](https://www.cnblogs.com/rickiyang/p/11074180.html)**

Go原生提供了连接数据库操作的支持，在用 Golang进行开发的时候，如果需要在和数据库交互，则可以使用database/sql包。这是一个对关系型数据库的通用抽象，它提供了标准的、轻量的、面向行的接口。

在Go中访问数据库需要用到sql.DB接口：它可以创建语句(statement)和事务(transaction)，执行查询，获取结果。

使用数据库时，除了database/sql包本身，还需要引入想使用的特定数据库驱动。官方不提供实现，先下载第三方的实现，[点击这里](https://github.com/golang/go/wiki/SQLDrivers)查看各种各样的实现版本。

本文测试数据库为mysql，使用的驱动为:github.com/go-sql-driver/mysql,需要引入的包为：

**解释一下导入包名前面的"\_"作用：**

import 下划线（如：import \_ github/demo）的作用：当导入一个包时，该包下的文件里所有init()函数都会被执行，然而，有些时候我们并不需要把整个包都导入进来，仅仅是是希望它执行init()函数而已。这个时候就可以使用 import \_ 引用该包。

上面的mysql驱动中引入的就是mysql包中各个init()方法，你无法通过包名来调用包中的其他函数。导入时，驱动的初始化函数会调用sql.Register将自己注册在database/sql包的全局变量sql.drivers中，以便以后通过sql.Open访问。

该包提供了一些类型（概括性的），每个类型可能包括一个或多个概念。

* DB  
  sql.DB 类型代表了一个数据库。这点和很多其他语言不同，它并不代表一个到数据库的具体连接，而是一个能操作的数据库对象，具体的连接在内部通过连接池来管理，对外不暴露。这点是很多人容易误解的：每一次数据库操作，都产生一个 sql.DB 实例，操作完 Close。
* Results  
  定义了三种结果类型：sql.Rows、sql.Row 和 sql.Result，分别用于获取多个多行结果、一行结果和修改数据库影响的行数（或其返回 last insert id）。
* Statements  
  sql.Stmt 代表一个语句，如：DDL、DML 等

// mysql

**package** main

**import** (

"database/sql"

"fmt"

\_ "github.com/Go-SQL-Driver/MySQL"

)

**func** main() {

//连接至数据库

// sql.Open()中的数据库连接串格式为："用户名:密码@tcp(IP:端口)/数据库?charset=utf8"。

db, err := sql.Open("mysql", "root:@tcp(localhost:3306)/mysql")

CheckErr(err)

insert(db)

//关闭数据库连接

db.Close()

}

//检查错误信息

**func** CheckErr(err error) {

**if** err != nil {

panic(err)

}

}

//插入demo

**func** insert(db \*sql.DB) {

//准备插入操作

//

// db.

res, err := db.Exec("create database db3")

CheckErr(err)

//查询删除多少条信息

num, err := res.RowsAffected()

CheckErr(err)

fmt.Println(num) //1 if creted database ok ..

}

//查询操作

**func** Query(db \*sql.DB) {

rows, e := db.Query("select name from mysql.help\_topic where name >'a' limit 3 ")

CheckErr(e)

//cols, \_ := rows.Columns()

**for** rows.Next() {

**var** name string

rows.Scan(&name)

fmt.Println(name)

fmt.Println(777)

//fmt.Println(json.Marshal()

}

}

5500

38k+20k +2k

1207 Bls 2k pls shk bob

19k-7 12k