



链滴

# Go | 语言对 MySQL 数据库的增、删、改、查操作

作者: [haojieli](#)

原文链接: <https://ld246.com/article/1489540662342>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1、建立数据库名为: **go\_test\_db**

2、建表名为: **userinfo**

字段:

**uid** int

**username** varchar

**\*\*language \*\***varchar

**created** varchar

3、对数据库操作

```
package main
```

```
import (  
    "database/sql"  
    "fmt"
```

```
    _ "github.com/Go-SQL-Driver/MySQL"  
)
```

```
func main() {  
    db, err := sql.Open("mysql", "go:golang@/go_test_db?charset=utf8")  
    //在这里解析一下sql.Open传的参数解析：  
    //db,err := sql.Open("mysql","登录名:登录密码@/数据库名?charset=utf8")  
    checkErr(err)
```

```
    //插入数据
```

```
    stmt, err := db.Prepare("INSERT userinfo SET username=?,language=?,created=?")
```

```
    checkErr(err)
```

```
    res, err := stmt.Exec("haojieli", "Java", "2016-07-12")
```

```
    checkErr(err)
```

```
    id, err := res.LastInsertId()
```

```
    checkErr(err)
```

```
    fmt.Println("获取添加成功的 value=", id)<br>
```

```
    //更新数据
```

```
    stmt, err = db.Prepare("update userinfo set username=? where uid=?")
```

```
    checkErr(err)
```

```
    res, err = stmt.Exec("update later", id)
```

```
    checkErr(err)
```

```
    affect, err := res.RowsAffected()
```

```
    checkErr(err)
```

```
    fmt.Println("更新当前添加的数据，为1就是更新成功. value =", affect)
```

```
    //查询数据
```

```
    rows, err := db.Query("SELECT * FROM userinfo")//本例子是查询userinfo表所有的数据，分  
改变Mysql语句就可以了，分页语句例子：select * from userinfo limit 0,5 (limit解析查看另外一  
文章)
```

```
    checkErr(err)
```

```
    fmt.Println("查询所有的userinfo----- start")
```

```

for rows.Next() {
    var uid int
    var username string
    var department string
    var created string
    err = rows.Scan(&uid, &username, &language, &created)
    checkErr(err)
    fmt.Print(uid, "|")
    fmt.Print(username, "|")
    fmt.Print(language, "|")
    fmt.Println(created)

}
fmt.Println("查询所有的userinfo----- end")<br>
//删除数据
stmt, err = db.Prepare("delete from userinfo where uid=?")
checkErr(err)
res, err = stmt.Exec(id - 1)
checkErr(err)
fmt.Println("删除添加id的上一条数据，为1就是成功删除. value =", affect)
db.Close()
}<br><br>
//检查是否有错
func checkErr(err error) {
    if err != nil {
        panic(err)//panic是获取到真正意义上的错误
    }
}

```

博文到此结束，感谢您的观看，希望对各位读者有所帮助，如果有什么意见以及建议请在评论留言.....

=====祝各位读者生活愉快=====

我的原文地址：<http://www.cnblogs.com/haojieli/p/5665869.html>