

ORM-

Fields/FieldsEx

- 1. Fields
- 2. FieldsEx

Fields

- 1. user4uid, nickname, passport, password
- 2.

```
// SELECT `uid`,`nickname` FROM `user` ORDER BY `uid` asc
db Table("user").Fields("uid, nickname").Order("uid asc").All()
```
- 3.

```
m := g Map{
    "uid"      : 10000
    "nickname" : "John Guo",
    "passport"  : "john",
    "password"  : "123456",
}

db Table table.Fields("nickname,passport,password").Data m.Insert()
// INSERT INTO `user`(`nickname`,`passport`,`password`) VALUES('John
Guo','john','123456')
```

FieldsEx

- 1. user4uid, nickname, passport, password
- 2.

```
// SELECT `uid`,`nickname` FROM `user`
db Table("user").FieldsEx "passport, password").All()
```
- 3.

```
m := g Map{
    "uid"      : 10000
    "nickname" : "John Guo",
    "passport"  : "john",
    "password"  : "123456",
}

db Table table.FieldsEx("uid").Data m.Insert()
// INSERT INTO `user`(`nickname`,`passport`,`password`) VALUES('John
Guo','john','123456')
```

Filter

```
gdb().Filter/map/struct/[]map/[]string
user4uid, nickname, passport, password

r, err := db Table("user").Filter().Data(g Map{
    "id"      : 1,
    "uid"      : 1,
    "passport"  : "john",
    "password"  : "123456",
}) Insert()
// INSERT INTO user(uid,passport,password) VALUES(1, "john", "123456")
```

~~isSQL~~



DataFilter/



GoFrame v1.15.7filterFilter

Content Menu

- Fields/FieldsEx
 - Fields
 - FieldsEx
- Filter
- OmitEmpty
 - /
 -

OmitEmpty

/

/Insert, Replace, Update, Save

```
map/struct nil,"",0 gdbmapstruct

// UPDATE `user` SET `name`='john',update_time=null WHERE `id`=1
db Table("user").Data g Map
    "name"      : "john",
    "update_time" : nil
}) Where("id", 1) Update()
```

OmitEmpty

```
// UPDATE `user` SET `name`='john' WHERE `id`=1
db Table("user").OmitEmpty().Data g Map
    "name"      : "john",
    "update_time" : nil
}) Where("id", 1) Update()
```

struct

```
type User struct {
    Id          int    `orm:"id"`
    Passport    string  `orm:"passport"`
    Password    string  `orm:"password"`
    NickName    string  `orm:"nickname"`
    CreateTime  string  `orm:"create_time"`
    UpdateTime  string  `orm:"update_time"`
}

user := User{
    Id          : 1
    NickName    : "john",
    UpdateTime  : gtime.Now().String(),
}

db Table("user").OmitEmpty().Data user.Insert()
// INSERT INTO `user`(`id`,`nickname`,`update_time`) VALUES(1,'john','2019-10-01 12:00:00')
```



/OmitEmpty

omitemptyOmitEmpty

1. structomitemptyjsonORMstruct
2. omitemptyOmitEmptyORMstructomitemptyOmitEmptystructOmitEmptystruct

whereOmitEmpty

```
// SELECT * FROM `user` WHERE `passport`='john' LIMIT 1
r, err := db Table("user").Where g Map{
    "nickname" : ""
    "passport" : "john"
}).OmitEmpty().One()
```

```
type User struct {
    Id          int    `orm:"id"`
    Passport    string  `orm:"passport"`
    Password    string  `orm:"password"`
    NickName    string  `orm:"nickname"`
    CreateTime  string  `orm:"create_time"`
    UpdateTime  string  `orm:"update_time"`
}

user := User{
    Passport : "john",
}

r, err := db Table("user").OmitEmpty().Where user.One()
// SELECT * FROM `user` WHERE `passport`='john' LIMIT 1
```