

ORM-

Content Menu

-
- [Record](#)
- [Result](#)

```
type Value = *gvar.Var //
type Record map[string]Value //
type Result []Record //
```

1. Value/Record/ResultORM
2. ResultRecordValue
3. Value*gvar.Var

```
mysql> select * from user;
+----+-----+-----+-----+-----+-----+
| id | passport | password | nickname | create_at | update_at |
+----+-----+-----+-----+-----+-----+
| 1 | john | sso5uf*654gf | John Great | 2023-10-01 12:00:00 | 2023-10-01 12:00:00 |
| 2 | smith | 0vKd3cjgffjh | Kyler Smith | 2023-10-01 12:10:00 | 2023-10-01 13:56:12 |
| 3 | jim | 3sxJWfySFjHIZqbnIAFC | Jim Green | 2023-10-01 12:10:00 | 2023-10-01 12:10:00 |
| 4 | sim | sfsdfowleu*76v | Sim Flangan | 2023-10-01 12:10:00 | 2023-10-01 12:10:00 |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

```
mysql> select * from user;
+----+-----+-----+-----+-----+-----+
| id | passport | password | nickname | create_at | update_at |
+----+-----+-----+-----+-----+-----+
| 1 | john | sso5uf*654gf | John Great | 2023-10-01 12:00:00 | 2023-10-01 12:00:00 |
| 2 | smith | 0vKd3cjgffjh | Kyler Smith | 2023-10-01 12:10:00 | 2023-10-01 13:56:12 |
| 3 | jim | 3sxJWfySFjHIZqbnIAFC | Jim Green | 2023-10-01 12:10:00 | 2023-10-01 12:10:00 |
| 4 | sim | sfsdfowleu*76v | Sim Flangan | 2023-10-01 12:10:00 | 2023-10-01 12:10:00 |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

```
mysql> select * from user;
+----+-----+-----+-----+-----+-----+
| id | passport | password | nickname | create_at | update_at |
+----+-----+-----+-----+-----+-----+
| 1 | john | sso5uf*654gf | John Great | 2023-10-01 12:00:00 | 2023-10-01 12:00:00 |
| 2 | smith | 0vKd3cjgffjh | Kyler Smith | 2023-10-01 12:10:00 | 2023-10-01 13:56:12 |
| 3 | jim | 3sxJWfySFjHIZqbnIAFC | Jim Green | 2023-10-01 12:10:00 | 2023-10-01 12:10:00 |
| 4 | sim | sfsdfowleu*76v | Sim Flangan | 2023-10-01 12:10:00 | 2023-10-01 12:10:00 |
+----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

Record

<https://pkg.go.dev/github.com/gogf/gf/v2/database/gdb>

`gdbmap/struct`

```
CREATE TABLE `user` (
  `uid` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(30) NOT NULL DEFAULT '' COMMENT '',
  `site` varchar(255) NOT NULL DEFAULT '' COMMENT '',
  PRIMARY KEY (`uid`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

```
uid name site
1 john https://goframe.org
```

```

package main

import (
    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

type User struct {
    Uid int
    Name string
}

func main() {
    var (
        user *User
        ctx = gctx.New()
    )
    err := g.DB().Model("user").Where("uid", 1).Scan(&user)
    if err != nil {
        g.Log().Header(false).Fatal(ctx, err)
    }
    if user != nil {
        g.Log().Header(false).Print(ctx, user)
    }
}

```

```
{ "Uid":1, "Name": "john" }
```

structUidNameORM

`gdb.Model.Scanstructstruct&user**User[]*User`

mapuid,name,sitestructUid,Name

1. struct()
2. -/_/ struct
- 3.

```

    struct
name      Name      match
Email     Email     match
nickname  NickName  match
NICKNAME  NickName  match
Nick-Name NickName  match
nick_name NickName  match
nick_name Nick_Name match
NickName  Nick_Name match
Nick-Name Nick_Name match

```



structgconv.Struct -Struct

Result

Result/RecordMap/ListJSON/XML

<https://pkg.go.dev/github.com/gogf/gf/v2/database/gdb>

`Record.MapResult.ListORM[]byteValue -gvar ResultRecordjsonxml`

```

package main

import (
    "database/sql"

    "github.com/gogf/gf/v2/frame/g"
    "github.com/gogf/gf/v2/os/gctx"
)

type User struct {
    Uid int
    Name string
    Site string
}

func main() {
    var (
        user []*User
        ctx = gctx.New()
    )
    err := g.DB().Model("user").Where("uid", 1).Scan(&user)
    if err != nil && err != sql.ErrNoRows {
        g.Log().Header(false).Fatal(ctx, err)
    }
    if user != nil {
        g.Log().Header(false).Print(ctx, user)
    }
}

```

```
[{"Uid":1,"Name":"john","Site":"https://goframe.org"}]
```