

ORM-Hook

HookModelCURD

Content Menu

•
•

Hook

```
type (  
    HookFuncSelect func(ctx context.Context, in *HookSelectInput)  
(result Result, err error)  
    HookFuncInsert func(ctx context.Context, in *HookInsertInput)  
(result sql.Result, err error)  
    HookFuncUpdate func(ctx context.Context, in *HookUpdateInput)  
(result sql.Result, err error)  
    HookFuncDelete func(ctx context.Context, in *HookDeleteInput)  
(result sql.Result, err error)  
)  
  
// HookHandler manages all supported hook functions for Model.  
type HookHandler struct {  
    Select HookFuncSelect  
    Insert HookFuncInsert  
    Update HookFuncUpdate  
    Delete HookFuncDelete  
}
```

Hook

```
// Hook sets the hook functions for current model.  
func (m *Model) Hook(hook HookHandler) *Model
```

birth_day

```
// Hook function definition.  
var hook = gdb.HookHandler{  
    Select: func(ctx context.Context, in *gdb.HookSelectInput) (result  
gdb.Result, err error) {  
        result, err = in.Next(ctx)  
        if err != nil {  
            return  
        }  
        for i, record := range result {  
            if !record["birth_day"].IsEmpty() {  
                age := gtime.Now().Sub(record["birth_day"].  
GTime()).Hours() / 24 / 365  
                record["age"] = gvar.New(age)  
            }  
            result[i] = record  
        }  
        return  
    },  
}  
  
// It registers hook function, creates and returns a new `model`.  
model := g.Model("user").Hook(hook)  
  
// The hook function takes effect on each ORM operation when using the  
`model`.  
all, err := model.Where("status", "online").OrderAsc(`id`).All()
```

