

ORM-AllAndCount

v2.5.0

```
// AllAndCount retrieves all records and the total count of records from the model.
// If useFieldForCount is true, it will use the fields specified in the model for counting;
// otherwise, it will use a constant value of 1 for counting.
// It returns the result as a slice of records, the total count of records, and an error if any.
// The where parameter is an optional list of conditions to use when retrieving records.
//
// Example:
//
//      var model Model
//      var result Result
//      var count int
//      where := []interface{}{"name = ?", "John"}
//      result, count, err := model.AllAndCount(true)
//      if err != nil {
//          // Handle error.
//      }
//      fmt.Println(result, count)
func (m *Model) AllAndCount(useFieldForCount bool) (result Result, totalCount int, err error)
```

Limit/Page

```
// SELECT `uid`,`name` FROM `user` WHERE `status`='deleted' LIMIT 0,10
// SELECT COUNT(`uid`,`name`) FROM `user` WHERE `status`='deleted'
all, count, err := Model("user").Fields("uid", "name").Where("status", "deleted").Limit(0, 10).AllAndCount(true)

// SELECT `uid`,`name` FROM `user` WHERE `status`='deleted' LIMIT 0,10
// SELECT COUNT(1) FROM `user` WHERE `status`='deleted'
all, count, err := Model("user").Fields("uid", "name").Where("status", "deleted").Limit(0, 10).AllAndCount(false)
```