ORM

goframeORM

1. *gdb.TX
2.

https://godoc.org/github.com/gogf/gf/database/gdb#TX

Begin/Commit/Rollback

db.Begin*gdb.Txtx.Committx.Rollback

⚠ Commit/RollbackdefergoroutineTransaction

1.

```
if tx, err := db.Begin(); err == nil {
    fmt.Println("")
}
```

db API

2.

```
if tx, err := db.Begin(); err == nil {
    r, err := tx.Save("user", gdb.Map{
        "id"   :  1,
        "name" : "john",
    })
    tx.Rollback()
    fmt.Println(r, err)
}
```

3.

```
if tx, err := db.Begin(); err == nil {
    r, err := tx.Save("user", gdb.Map{
        "id"   :  1,
        "name" : "john",
    })
    tx.Commit()
    fmt.Println(r, err)
}
```

4.

tx.Tabletx.Fromdb.Tabledb.From

```
if tx, err := db.Begin(); err == nil {
    r, err := tx.Table("user").Data(gdb.Map{"id":1, "name": "john_1"}).Save()
    tx.Commit()
    fmt.Println(r, err)
}
```

ORM()

Transaction

gdbTransaction

```
func (db DB) Transaction(f func(tx *TX) error) (err error)
```

errornilCommitRollback

✓ panic

```
db.Transaction(func(tx *gdb.TX) error {
    // user
    result, err := tx.Insert("user", g.Map{
        "passport": "john",
        "password": "12345678",
        "nickname": "JohnGuo",
    })
    if err != nil {
        return err
    }
    // user_detail
    id, err := result.LastInsertId()
    if err != nil {
        return err
    }
    _, err = tx.Insert("user_detail", g.Map{
        "uid":       id,
        "site":      "https://johng.cn",
        "true_name": "GuoQiang",
    })
    if err != nil {
        return err
    }
    return nil
})
```

GoFramev1.15.7ORMTransaction Save Point

```
// Begin starts a nested transaction procedure.
func (tx *TX) Begin() error

// Commit commits current transaction.
// Note that it releases previous saved transaction point if it's in a
nested transaction procedure,
// or else it commits the hole transaction.
func (tx *TX) Commit() error

// Rollback aborts current transaction.
// Note that it aborts current transaction if it's in a nested transaction
procedure,
// or else it aborts the hole transaction.
func (tx *TX) Rollback() error

// SavePoint performs `SAVEPOINT xxx` SQL statement that saves transaction
at current point.
// The parameter `point` specifies the point name that will be saved to
server.
func (tx *TX) SavePoint(point string) error

// RollbackTo performs `ROLLBACK TO SAVEPOINT xxx` SQL statement that
rollbacks to specified saved transaction.
// The parameter `point` specifies the point name that was saved
previously.
func (tx *TX) RollbackTo(point string) error

// Transaction wraps the transaction logic using function `f`.
// It rollbacks the transaction and returns the error from function `f` if
// it returns non-nil error. It commits the transaction and returns nil if
// function `f` returns nil.
//
// Note that, you should not Commit or Rollback the transaction in
function `f`
// as it is automatically handled by this function.
func (tx *TX) Transaction(f func(tx *TX) error) (err error)
```

1.

SQLidname

```
CREATE TABLE `user` (
  `id` int(10) unsigned NOT NULL COMMENT 'ID',
  `name` varchar(45) NOT NULL COMMENT '',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
tx, err := db.Begin()
if err != nil {
        panic(err)
}
if err = tx.Begin(); err != nil {
        panic(err)
}
_, err = tx.Model(table).Data(g.Map{"id": 1, "name": "john"}).Insert()
if err = tx.Rollback(); err != nil {
        panic(err)
}
_, err = tx.Model(table).Data(g.Map{"id": 2, "name": "smith"}).Insert()
if err = tx.Commit(); err != nil {
        panic(err)
}
```

db.Begintx.Begin

db.Begintx.Begindb.Begintxtxtx.BeginSavePointtransactionNNSAVEPOINT `transaction1`20

goframeORMSQL

```
2021-05-02 13:40:15.483 [DEBU] [  0 ms] [default] SAVEPOINT `transaction0`
2021-05-02 13:40:15.485 [DEBU] [  2 ms] [default] SHOW FULL COLUMNS FROM
`user`
2021-05-02 13:40:15.486 [DEBU] [  0 ms] [default] INSERT INTO `user`(`id`,
`name`) VALUES(1,'john')
2021-05-02 13:40:15.486 [DEBU] [  0 ms] [default] ROLLBACK TO SAVEPOINT
`transaction0`
2021-05-02 13:40:15.486 [DEBU] [  0 ms] [default] INSERT INTO `user`(`id`,
`name`) VALUES(2,'smith')
2021-05-02 13:40:15.487 [DEBU] [  1 ms] [default] COMMIT
```

```
mysql> select * from `user`;
+----+-------+
| id | name  |
+----+-------+
|  2 | smith |
+----+-------+
1 row in set (0.00 sec)
```

2.

Transaction

Transaction

```
if err = db.Transaction(func(tx *gdb.TX) error {
        // Nested transaction 1.
        if err = tx.Transaction(func(tx *gdb.TX) error {
                _, err = tx.Model(table).Data(g.Map{"id": 1, "name":
"john"}).Insert()
                return err
        }); err != nil {
                return err
        }
        // Nested transaction 2, panic.
        if err = tx.Transaction(func(tx *gdb.TX) error {
                _, err = tx.Model(table).Data(g.Map{"id": 2, "name":
"smith"}).Insert()
                // Create a panic that can make this transaction rollback
automatically.
                panic("error")
        }); err != nil {
                return err
        }
        return nil
}); err != nil {
        panic(err)
}
```

SQL

```
2021-05-02 13:42:01.935 [DEBU] [  1 ms] [default] SAVEPOINT `transaction0`
2021-05-02 13:42:01.939 [DEBU] [  4 ms] [default] SHOW FULL COLUMNS FROM
`user`
2021-05-02 13:42:01.940 [DEBU] [  0 ms] [default] INSERT INTO `user`(`id`,
`name`) VALUES(1,'john')
2021-05-02 13:42:01.940 [DEBU] [  0 ms] [default] RELEASE SAVEPOINT
`transaction0`
2021-05-02 13:42:01.940 [DEBU] [  0 ms] [default] SAVEPOINT `transaction0`
2021-05-02 13:42:01.941 [DEBU] [  0 ms] [default] INSERT INTO `user`(`id`,
`name`) VALUES(2,'smith')
2021-05-02 13:42:01.941 [DEBU] [  0 ms] [default] ROLLBACK TO SAVEPOINT
`transaction0`
2021-05-02 13:42:01.941 [DEBU] [  0 ms] [default] ROLLBACK
```

3. SavePoint/RollbackTo

Transaction Save PointSavePointPoint

```
tx, err := db.Begin()
if err != nil {
        panic(err)
}
defer func() {
        if err := recover(); err != nil {
                _ = tx.Rollback()
        }
}()
if _, err = tx.Model(table).Data(g.Map{"id": 1, "name": "john"}).Insert();
err != nil {
        panic(err)
}
if err = tx.SavePoint("MyPoint"); err != nil {
        panic(err)
}
if _, err = tx.Model(table).Data(g.Map{"id": 2, "name": "smith"}).Insert();
err != nil {
        panic(err)
}
if _, err = tx.Model(table).Data(g.Map{"id": 3, "name": "green"}).Insert();
err != nil {
        panic(err)
}
if err = tx.RollbackTo("MyPoint"); err != nil {
        panic(err)
}
if err = tx.Commit(); err != nil {
        panic(err)
}
```

SQL

```
2021-05-02 13:59:36.788 [DEBU] [  3 ms] [default] SHOW FULL COLUMNS FROM
`user`
2021-05-02 13:59:36.788 [DEBU] [  0 ms] [default] INSERT INTO `user`
(`name`,`id`) VALUES('john',1)
2021-05-02 13:59:36.789 [DEBU] [  1 ms] [default] SAVEPOINT `MyPoint`
2021-05-02 13:59:36.789 [DEBU] [  0 ms] [default] INSERT INTO `user`(`id`,
`name`) VALUES(2,'smith')
2021-05-02 13:59:36.789 [DEBU] [  0 ms] [default] INSERT INTO `user`
(`name`,`id`) VALUES('green',3)
2021-05-02 13:59:36.789 [DEBU] [  0 ms] [default] ROLLBACK TO SAVEPOINT
`MyPoint`
2021-05-02 13:59:36.791 [DEBU] [  2 ms] [default] COMMIT
```

```
mysql> select * from `user`;
+----+------+
| id | name |
+----+------+
|  1 | john |
+----+------+
1 row in set (0.00 sec)
```

InsertSavePointMyPointRollbackToInsert