## 1、. “xinda.im/boss/models”

表示可以引用models包中的全局变量，函数，不用包命.

## 2、sql：

（1）除了查询操作，其他操作需要用事务。

（2）rows.Scan

这个函数的参数个数必须和数据库表字段的个数一样

1. 别的表对于本表的依赖删除了，本表的记录才能删除

res, err := stmt.Exec(employee.Name, employee.Sex, employee.Email, employee.Phone)

    beego.Info(res.LastInsertId())

当Id为自动增长的，就可以通过lastId, err := res.LastInsertId()

获取插入后的Id

1. mysql不同的表的字段名要不一样，防止
2. 字段类型是字符串的话，默认要default '' 否则，可能插入的时候没有值，拿出来就是null的，会报错
3. 事务：

事务是原子的，在每一次Exec执行过程中，如果出错，则可以回退，最后都没有出错，才进行提交到数据库（commit）

事物步骤：

1. 开始事务：
2. 执行操作a,b,c每次执行都用同一个错误对象err接收错误结果
3. 执行错误检查，有错误就回滚，没有错误就把刚刚修改的东西提交到数据库(如图)

defer func() {

        if err == nil {

            tx.Commit()

        } else {

            tx.Rollback()

        }

    }()

1. 善用主键对应的记录存在时，更新记录，否则，插入记录语句：

insert into agent\_customer (AgentId,CustomerId) values (?,?) on duplicate key update AgentId = ?

比如，这条语句中，CustomerId是主键，当CustomerId这个主键的记录存在时，更新update后面的数据，否则，插入一条新的纪录。

（9）Scan的值为null的问题？ 如果存入数据库的数据为null，然后Scan接收的字段不为指针类型，就不能接收该null。因此解决的方法就是把字段弄成指针类型。

## 模版：

## select：

1. select的功能：监听I/O操作，当I/O操作发生时，触发相应的动作。
2. select的代码形式和switch非常相似，不过select的case里的操作语句只能是I/O操作
3. <http://blog.csdn.net/zhonglinzhang/article/details/45913443>
4. 实例1：

for {

            select {

            case <-time.After(1 \* time.Hour):

                this.\_load()

            }

        }

1. 表示多少时间之后，但是在取出channel内容之前不阻塞，后续程序可以继续执行。参见：<http://studygolang.com/articles/4331>
2. 由于select等待time.After的I/O操作，因此相当于阻塞了。因此达到1小时以后执行this.\_load()语句的效果。