Meetup #9

SQL & ORM
In Golang

Who am I

- Triet Pham
- Backend developer @ Lozi.vn
- triet.phm@gmail.com

Agenda

1. Package SQL & Database driver
2. ORM
3. Migration

Package SQL

Package sql provides a generic interface around SQL (or SQL-like) databases. The sql package must be used in conjunction with a database driver.

https://golang.org/pkg/database/sql/

Database driver

The database/sql and database/sql/driver packages are designed for using databases from Go and implementing database drivers, respectively.

https://github.com/golang/go/wiki/SQLDrivers

Database driver

- Couchbase N1QL: https://github.com/couchbase/go_n1ql
- MS SQL Server (pure go): https://github.com/denisenkom/go-mssqldb
- MySQL: https://github.com/go-sql-driver/mysql/
- Oracle: https://github.com/mattn/go-oci8
- Postgres (pure Go): https://github.com/lib/pq
- SQLite: https://github.com/mattn/go-sqlite3
- DB2: https://bitbucket.org/phiggins/db2cli
- ODBC: https://github.com/alexbrainman/odbc
- ...

Package SQL

```
type DB struct {
    // contains filtered or unexported fields
}
```

- Representing a pool of zero or more underlying connections.
- Safe for concurrent use by multiple goroutines.
- Creates and frees connections automatically, also maintains a free pool of idle connections.
- SetConnMaxLifetime, SetMaxIdleConns, SetMaxOpenConns

Package SQL

```
type Stmt struct {
    // contains filtered or unexported fields
}
```

Stmt is a prepared statement. A Stmt is safe for concurrent use by multiple goroutines.

- func Close() error
- func Exec(args ...interface{}) (Result, error)
- func Query(args ...interface{}) (*Rows, error)
- func QueryRow(args ...interface{}) *Row

Package SQL

```
type Rows struct {
    // contains filtered or unexported fields
}
```

Rows is the result of a query. Its cursor starts before the first row of the result set

- func Close() error
- func Columns() ([]string, error)
- func Err() error
- func Next() bool
- func Scan(dest ...interface{}) error

Package SQL

```
type Tx struct {
    // contains filtered or unexported fields
}
```

Tx is an in-progress database transaction. A transaction must end with a call to Commit or Rollback.

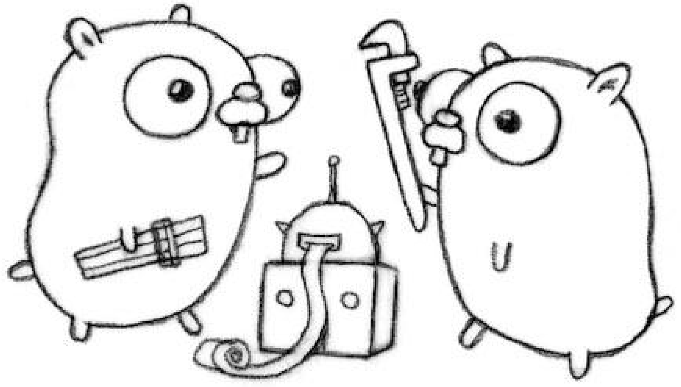
func Commit() error

func Exec(query string, args ...interface{}) (Result, error)

func Query(query string, args ...interface{}) (*Rows, error)

func Rollback() error

Package SQL



DEMO

ORM

Object-relational mapping (**ORM**, **O/RM**, and **O/R mapping tool**) in computer science is a programming technique for converting data between incompatible type systems in object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language *(Wikipedia - https://en.wikipedia.org/wiki/Object-relational_mapping)*

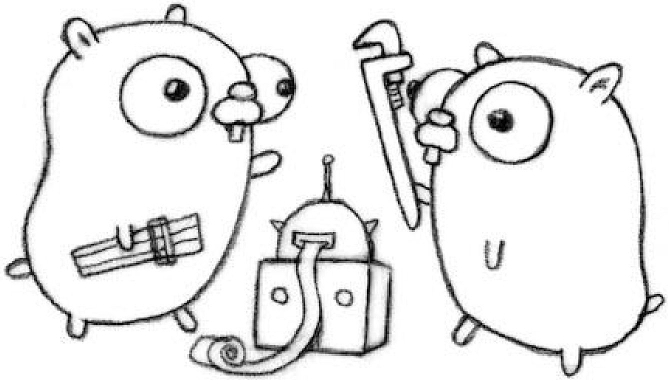
ORM

- beego orm - A powerful orm framework for go. Support: pq/mysql/sqlite3.
- GORM - The fantastic ORM library for Golang, aims to be developer friendly.
- gorp - Go Relational Persistence, ORM-ish library for Go.
- reform - A better ORM for Go, based on non-empty interfaces and code generation.
- Xorm - Simple and powerful ORM for Go.

ORM - Gorm

- Full-Featured ORM (almost)
- Associations (Has One, Has Many, Belongs To, Many To Many, Polymorphism)
- Callbacks (Before/After Create/Save/Update/Delete/Find)
- Preloading (eager loading)
- Transactions
- Composite Primary Key
- SQL Builder
- Auto Migrations
- Logger
- Extendable, write Plugins based on GORM callbacks

ORM



DEMO

ORM should or shouldn’t?

Advantages:

- Simplicity
- Code generation
- Efficiency is good enough in the early stage
- Protect from SQL Injection
- Easier for maintenance

ORM should or shouldn’t?

Disadvantage:

- Have to learn the ORM framework first. There is a learning curve in understanding and using ORM framework.
- Hundreds or thousands queries are executed behind
- Usually have to write your own SQL query for better performance

Migration

In software engineering, **schema migration** (also **database migration**, **database change management**) refers to the management of incremental, reversible changes to relational database schemas. A schema migration is performed on a database whenever it is necessary to update or revert that database's schema to some newer or older version.

(Wikipedia - https://en.wikipedia.org/wiki/Schema_migration)

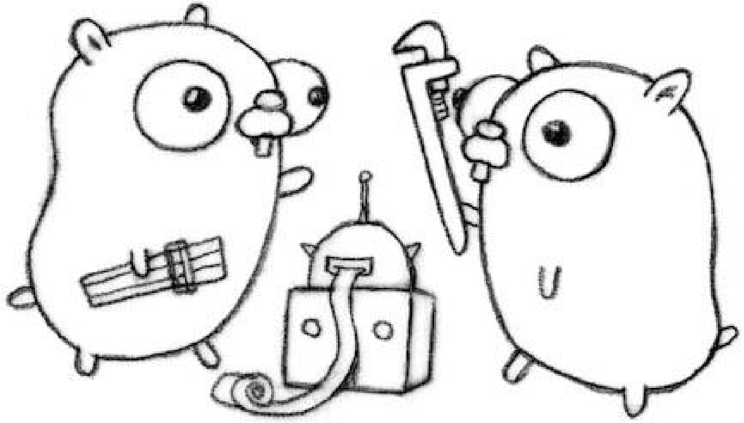
Migration

- Data models no longer need to be fully designed up-front. Allows for fixing mistakes and adapting the data as requirements change
- Database schemas versioning
- Supposing that the software under development interacts with a database, every version of the source code can be associated with at least one database schema with which it is compatible.
- Easier to test, if everything else fails, the amount of data is small enough for a human to process.

Migration

- Active Record (Migrations) - Ruby
- Alembic - Python
- Ruckusing-migration - PHP
- Goose - Go
- Ecto.Migrations - Elixir

ORM



DEMO

Q&A

recruitment
LOZI NEED YOU!