goose Database migrations

CONCEPTS

Set GOOSE_DRIVER, GOOSE_DBSTRING and GOOSE_MIGRATION_DIR (or a .env file) and drop the driver and dbstring from every command.

create new .sql file
↓ edit
Up / Down SQL annotations
↓ goose up
Database goose_db_version
↓ goose status
Verified applied list

CONNECT

goose *<driver>* *<dbstring>* *<command>*
Full invocation form

goose -dir *<path>* *<command>*
Migration directory (default .)

export GOOSE_DRIVER=*<driver>*
Driver from environment

export GOOSE_DBSTRING=*<dbstring>*
Connection string from environment

DBSTRING EXAMPLES

Driver and connection string

driver	dbstring
sqlite3	./app.db
postgres	"user=postgres dbname=app sslmode=disable"
mysql	"user:pass@/app?parseTime=true"
mssql	"sqlserver://user:pass@host:1433?database=app"
clickhouse	"tcp://127.0.0.1:9000"
redshift	"postgres://user:pass@host:5439/db"

CREATE

goose create *<name>* **sql**
New SQL migration

CREATE (cont.)

goose -dir *<path>* **create** *<name>* **sql**
Create in a given directory

goose validate
Check files without running them

SQL ANNOTATIONS

-- +goose Up
Start of forward statements

-- +goose Down
Start of rollback statements

-- +goose NO TRANSACTION
Run outside a transaction (file top)

-- +goose StatementBegin
Open a block with inner semicolons

MIGRATE

goose up
Apply all pending migrations

goose up-by-one
Apply just the next one

goose up-to *<version>*
Migrate up to a version

goose down
Roll back the latest one

goose down-to *<version>*
Roll back to a version

goose redo
Re-run the latest migration

INSPECT

goose status
List migrations and applied time

goose version
Print current DB version

INSTALL

go install github.com/pressly/goose/v3/cmd/goose@latest
Install with Go