

# The Go Programming Language

## Database

Anuchit Prasertsang  
Developer

# Database service

[www.elephantsql.com/](https://www.elephantsql.com/) (<https://www.elephantsql.com/>)

2

# Connect database

```
package main

import (
    "database/sql"
    "fmt"
    "log"
    "os"

    _ "github.com/lib/pq"
)

func main() {
    //db, err := sql.Open("postgres", "root:password@tcp(127.0.0.1:3306)/")
    db, err := sql.Open("postgres", os.Getenv("DATABASE_URL"))
    if err != nil {
        log.Fatal("Connect to database error", err)
    }
    defer db.Close()

    fmt.Println("okay")
}
```

3

# CREATE TABLE

```
func main() {
    db, err := sql.Open("postgres", os.Getenv("DATABASE_URL"))
    if err != nil {
        log.Fatal("Connect to database error", err)
    }
    defer db.Close()

    createTb := `
CREATE TABLE IF NOT EXISTS todos (
    id SERIAL PRIMARY KEY,
    title TEXT,
    status TEXT
);
`

    _, err = db.Exec(createTb)

    if err != nil {
        log.Fatal("can't create table", err)
    }

    fmt.Println("create table success")
}
```

4

# INSERT

```
func main() {
    db, err := sql.Open("postgres", os.Getenv("DATABASE_URL"))
    if err != nil {
        log.Fatal("Connect to database error", err)
    }
    defer db.Close()

    row := db.QueryRow("INSERT INTO todos (title, status) values (
var id int
err = row.Scan(&id)
if err != nil {
    fmt.Println("can't scan id", err)
    return
}

    fmt.Println("insert todo success id : ", id)
}
```

5

## QUERY one row

```
func main() {
    db, err := sql.Open("postgres", os.Getenv("DATABASE_URL"))
    if err != nil {
        log.Fatal("Connect to database error", err)
    }
    defer db.Close()

    stmt, err := db.Prepare("SELECT id, title, status FROM todos w
    if err != nil {
        log.Fatal("can't prepare query one row statement", err)
    }

    rowId := 1
    row := stmt.QueryRow(rowId)
    var id int
    var title, status string

    err = row.Scan(&id, &title, &status)
    if err != nil {
        log.Fatal("can't Scan row into variables", err)
    }

    fmt.Println("one row", id, title, status)
}
```

6

# UPDATE

```
func main() {  
    db, err := sql.Open("postgres", os.Getenv("DATABASE_URL"))  
    if err != nil {  
        log.Fatal("Connect to database error", err)  
    }  
    defer db.Close()  
  
    stmt, err := db.Prepare("UPDATE todos SET status=$2 WHERE id=$1")  
  
    if err != nil {  
        log.Fatal("can't prepare statment update", err)  
    }  
  
    if _, err := stmt.Exec(1, "inactive"); err != nil {  
        log.Fatal("error execute update ", err)  
    }  
  
    fmt.Println("update success")  
}
```

7

# QUERY all

```
stmt, err := db.Prepare("SELECT id, title, status FROM todos")
if err != nil {
    log.Fatal("can't prepare query all todos statment", err)
}

rows, err := stmt.Query()
if err != nil {
    log.Fatal("can't query all todos", err)
}
for rows.Next() {
    var id int
    var title, status string
    err := rows.Scan(&id, &title, &status)
    if err != nil {
        log.Fatal("can't Scan row into variable", err)
    }
    fmt.Println(id, title, status)
}

fmt.Println("query all todos success")
```

8



# Todo gin api + database

# workshop

# Thank you

Anuchit Prasertsang

Developer

[anuchit.prasertsang@gmail.com](mailto:anuchit.prasertsang@gmail.com) (mailto:anuchit.prasertsang@gmail.com)

<https://github.com/AnuchitO> (https://github.com/AnuchitO)

[@twitter\\_AnuchitO](http://twitter.com/twitter_AnuchitO) (http://twitter.com/twitter\_AnuchitO)