**Database operation**

Download postgresql

<https://www.enterprisedb.com/postgresql-tutorial-resources-training?uuid=db55e32d-e9f0-4d7c-9aef-b17d01210704&campaignId=7012J000001NhszQAC>

Download DBeaver

<https://dbeaver.io/download/>

Create a database called test in dbviewer

Code is as below & taken from

[https://att-c.udemycdn.com/2021-04-05\_23-12-41-ccb5b133039198cd672e083dafee1c72/original.zip?response-content-disposition=attachment%3B+filename%3Dtest\_connect.zip&Expires=1652886735&Signature=P-ADFLUqNG4i6xJIUwB2ukNITzHjJ3TTroisK2V7MOSsiKeC7eC7FxMw3dVCUEfeaMKi454dR~d~m~naWyZpuvFibWMU84GutwSxxIxpoDOg~EWY8lp~1l5ng6fJTHGKhu0kmKGdn8DJJJBDnNAlC5lfoGvVtsbg9VLnoHMPL24~56EEPjkqnNFteKOtm-GvSVaQz7lQZPwJQVnmCDRzv1oe0VaAozR4iKD1JQWgXYRER20ERT~50-5PycL73A~bj2rVh9qMVaOBqaFU1vLwaLTGguDm4LVV-jXSDI0blymO7mtRy4nW5QLOFC8gblmHhoOlL~UObnxe8o20uHtLhg\_\_&Key-Pair-Id=APKAITJV77WS5ZT7262A](https://att-c.udemycdn.com/2021-04-05_23-12-41-ccb5b133039198cd672e083dafee1c72/original.zip?response-content-disposition=attachment;+filename=test_connect.zip&Expires=1652886735&Signature=P-ADFLUqNG4i6xJIUwB2ukNITzHjJ3TTroisK2V7MOSsiKeC7eC7FxMw3dVCUEfeaMKi454dR~d~m~naWyZpuvFibWMU84GutwSxxIxpoDOg~EWY8lp~1l5ng6fJTHGKhu0kmKGdn8DJJJBDnNAlC5lfoGvVtsbg9VLnoHMPL24~56EEPjkqnNFteKOtm-GvSVaQz7lQZPwJQVnmCDRzv1oe0VaAozR4iKD1JQWgXYRER20ERT~50-5PycL73A~bj2rVh9qMVaOBqaFU1vLwaLTGguDm4LVV-jXSDI0blymO7mtRy4nW5QLOFC8gblmHhoOlL~UObnxe8o20uHtLhg__&Key-Pair-Id=APKAITJV77WS5ZT7262A)

package main

import (

    "database/sql"

    "fmt"

    "log"

    \_ "github.com/jackc/pgx/v4/stdlib"

)

func main() {

    // connect to a database

    conn, err := sql.Open("pgx", "host=localhost port=5432 dbname=test user=yagnik.pokal password=yagnik@2017")

    if err != nil {

        log.Fatal(fmt.Sprintf("Unable to connect: %v\n", err))

    }

    defer conn.Close()

    log.Println("Connected to database!")

    // test my connection

    err = conn.Ping()

    if err != nil {

        log.Fatal("Cannot ping database!")

    }

    log.Println("Pinged database!")

    // get rows from table

    err = getAllRows(conn)

    if err != nil {

        log.Fatal(err)

    }

    // insert a row

    query := `insert into users (first\_name, last\_name) values ($1, $2)`

    \_, err = conn.Exec(query, "Jack", "Brown")

    if err != nil {

        log.Fatal(err)

    }

    log.Println("Inserted a row!")

    // get rows from table again

    err = getAllRows(conn)

    if err != nil {

        log.Fatal(err)

    }

    // update a row

    stmt := `update users set first\_name = $1 where id = $2`

    \_, err = conn.Exec(stmt, "Jackie", 5)

    if err != nil {

        log.Fatal(err)

    }

    log.Println("Updated one or more rows")

    // get rows from table again

    err = getAllRows(conn)

    if err != nil {

        log.Fatal(err)

    }

    // get one row by id

    query = `select id, first\_name, last\_name from users where id = $1`

    var firstName, lastName string

    var id int

    row := conn.QueryRow(query, 1)

    err = row.Scan(&id, &firstName, &lastName)

    if err != nil {

        log.Fatal(err)

    }

    log.Println("QueryRow returns", id, firstName, lastName)

    // delete a row

    query = `delete from users where id = $1`

    \_, err = conn.Exec(query, 6)

    if err != nil {

        log.Fatal(err)

    }

    log.Println("Deleted a row!")

    // get rows from table again

    err = getAllRows(conn)

    if err != nil {

        log.Fatal(err)

    }

}

func getAllRows(conn \*sql.DB) error {

    rows, err := conn.Query("select id, first\_name, last\_name from users")

    if err != nil {

        log.Println(err)

        return err

    }

    defer rows.Close()

    var firstName, lastName string

    var id int

    for rows.Next() {

        err := rows.Scan(&id, &firstName, &lastName)

        if err != nil {

            log.Println(err)

            return err

        }

        fmt.Println("Record is", id, firstName, lastName)

    }

    if err = rows.Err(); err != nil {

        log.Fatal("Error scanning rows", err)

    }

    fmt.Println("---------------------------")

    return nil

}



