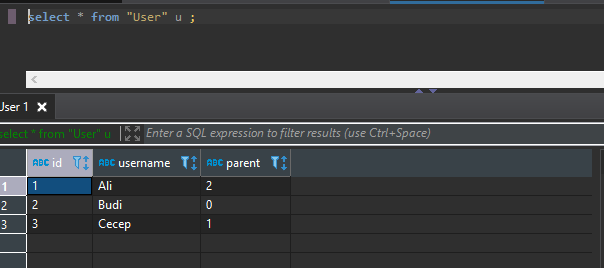
Name : Gilang Saputro

**Answer for number 1,3,4**

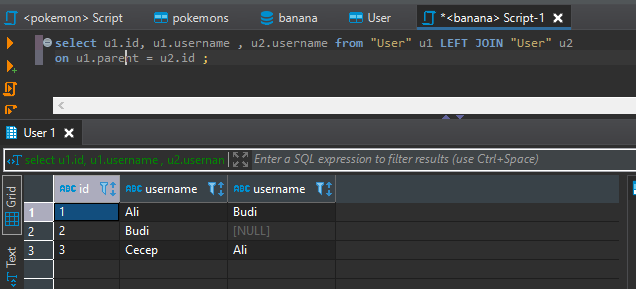
1. **Simple Database querying**



**Answer :**

**select** u1.id, u1.username , u2.username **from** "User" u1 **LEFT** **JOIN** "User" u2

**on** u1.parent = u2.id ;



1. **Answer in : https://github.com/derpl-del/simple\_api**

**Refactor the code**

**Answer :**

func findFirstStringInBracket(str string) string {

    pos1 := strings.Index(str, "(")

*if* pos1 < 0 {

*return* ""

    }

    pos1++

    pos2 := strings.Index(str[pos1:], ")")

*if* pos2 < 0 {

*return* ""

    }

*return* str[pos1 : pos1+pos2]

}



1. **Logic Test**

**Answer :**

package main

import (

    "fmt"

    "sort"

)

type RuneSlice []rune

func (p RuneSlice) Len() int           { *return* len(p) }

func (p RuneSlice) Less(i, j int) bool { *return* p[i] < p[j] }

func (p RuneSlice) Swap(i, j int)      { p[i], p[j] = p[j], p[i] }

type Result [][]string

func (r Result) Len() int           { *return* len(r) }

func (r Result) Less(i, j int) bool { *return* len(r[i]) > len(r[j]) }

func (r Result) Swap(i, j int)      { r[i], r[j] = r[j], r[i] }

func main() {

    input := []string{"kita", "atik", "tika", "aku", "kia", "makan", "kua"}

    temp\_input := make(map[string][]string)

*//Grouping*

*for* \_, x := *range* input {

        r := StringToRuneSlice(x)

        temp\_input[r] = append(temp\_input[r], x)

    }

    res := [][]string{}

*//Slice rune to SliceString*

*for* \_, x := *range* temp\_input {

        res = append(res, x)

    }

*//Sort by Count of array*

    sort.Sort(Result(res))

    fmt.Println(res)

}

func StringToRuneSlice(input string) string {

    var r []rune

*for* \_, x := *range* input {

        r = append(r, x)

    }

    sort.Sort(RuneSlice(r))

*return* string(r)

}

