nguyen minh quan

INTS  Tố Hữu

RamdaJs

Table of Contents

[1. RamdaJS Pipe and Compose 2](#_Toc143617475)

[a. Syntax compose() 2](#_Toc143617476)

[b. Syntax pipe() 2](#_Toc143617477)

[2. Link các ứng app cá nhân 3](#_Toc143617478)

[3. Làm bài tập MySql 3](#_Toc143617479)

# [RamdaJS Pipe and Compose](https://www.youtube.com/watch?v=myGSs8lu62M&ab_channel=ChristopherOkhravi)

## Syntax compose()

Về mặt syntax thì

(f <<< g)(x) = f(g(x))

Tương đương với:

R.compose(f, g)(x) = f(g(x))

const f = x => x +'?'

const g = x => x +'!'

console.log("R.compose(f , g) ==>", R.compose(f , g)("Hello"));

A white background with black text

Description automatically generated

Figure Kết quả compose của R

Với compose thì right most có thể có any arity

## Syntax pipe()

(f >>>g)(x) = f(g(x))

Tương đương với:

R.pipe(f, g)(x) = f(g(x))

const f = x => x +'?'

const g = x => x +'!'

console.log("R.pipe(f , g) ==>", R.pipe(f , g)("Hello"));

A screen shot of a computer

Description automatically generated

Figure Kết quả của pipe()

Với compose thì left most có thể có any arity

## Link các ứng app cá nhân

Project SCSS, react và chia trang responsive:

* Deploy: [Site overview | merry-banoffee-32a813 | Netlify](https://app.netlify.com/sites/merry-banoffee-32a813/overview)
* Code: [quannm97/work-out-app: Work out plans app with react (github.com)](https://github.com/quannm97/work-out-app)

Project React, Redux, CRUD:

* Deploy: [React App (transcendent-malasada-fbba95.netlify.app)](https://transcendent-malasada-fbba95.netlify.app/)
* Code: [quannm97/Movie-app: movie-app with middleware redux (github.com)](https://github.com/quannm97/Movie-app)

Project thiết kế các tính năng Pagination, Validate với Js thuần :

* Deploy: [MindCode - To Do List App (friendly-unicorn-04bb2e.netlify.app)](https://friendly-unicorn-04bb2e.netlify.app/)
* Code: [quannm97/todo-list-master: todo-list-master (github.com)](https://github.com/quannm97/todo-list-master)
* Project này HTML và CSS em clone, chỉ soạn logic cho file JS
* Có 1 số bugs ở feature pagination là khi kéo data về, data là 1 promise, nên pagination method cũng phải thành 1 async function. Tuy vậy không phải tất cả dữ liệu đều phải sử dụng async, nên hiện em chưa có hướng xử lý cho phù hợp. Tương tự với method Validation.

## Làm bài tập MySql

**Bài tập** 1: Tạo cơ sở dữ liệu

Hãy tạo một cơ sở dữ liệu có tên "QuanLyNhanVien" chứa bảng "NhanVien" với các trường: "id" (kiểu INT), "ho\_ten" (kiểu VARCHAR), "tuoi" (kiểu INT) và "chuc\_vu" (kiểu VARCHAR).

***Giải***

CREATE DATABASE QuanLyNhanVien;

USE QuanLyNhanVien;

CREATE TABLE NhanVien (

id INT,

ho\_ten VARCHAR(255),

tuoi INT,

chuc\_vu VARCHAR(255)

)

**Bài tập 2**: Thêm dữ liệu vào bảng

Thêm 5 bản ghi vào bảng "NhanVien" với thông tin nhân viên khác nhau như họ tên, tuổi và chức vụ.

***Giải***

SELECT \* FROM quanlynhanvien.nhanvien;

USE quanlynhanvien;

INSERT INTO nhanvien (id,ho\_ten, tuoi, chuc\_vu)

VALUES

(1, "Nguyen Van A", 10, "truong phong" ),

(2, "Nguyen Van B", 20, "pho phong" ),

(3, "Nguyen Van C", 30, "truong nhom" ),

(4, "Nguyen Van D", 40, "pho nhom" ),

(5, "Nguyen Van E", 50, "nhan vien" )

;

A screenshot of a computer

Description automatically generated

Figure Kết quả bài 2

Bài tập 3: Cập nhật dữ liệu

Cập nhật thông tin của một nhân viên trong bảng "NhanVien". Ví dụ: thay đổi chức vụ của nhân viên có id là 1 thành "Trưởng phòng".

***Giải***

SELECT id

FROM `quanlynhanvien`.`nhanvien`;

USE quanlynhanvien;

UPDATE nhanVien SET chuc\_vu = "Trưởng Phòng" WHERE id=1;

Bài tập 4: Xóa dữ liệu

Xóa một nhân viên khỏi bảng "NhanVien" dựa trên điều kiện nào đó. Ví dụ: xóa nhân viên có id là 2.

***Giải***

SELECT id

FROM `quanlynhanvien`.`nhanvien`;

USE quanlynhanvien;

DELETE FROM nhanVien WHERE id=2;

Bài tập 5: Tạo truy vấn đơn giản

Thực hiện truy vấn để lấy danh sách các nhân viên trong bảng "NhanVien".

***Giải***

SELECT \* FROM quanlynhanvien.nhanvien;

Bài tập 6: Truy vấn với điều kiện

Thực hiện truy vấn để lấy danh sách nhân viên có tuổi lớn hơn hoặc bằng 25 trong bảng "NhanVien".

***Giải***

SELECT \*

FROM `quanlynhanvien`.`nhanvien`

WHERE tuoi >= 25

;

Bài tập 7: Sắp xếp kết quả truy vấn

Thực hiện truy vấn để lấy danh sách nhân viên được sắp xếp theo tuổi tăng dần trong bảng "NhanVien".

***Giải***

SELECT \*

FROM `quanlynhanvien`.`nhanvien`

ORDER BY tuoi ASC

;

Bài tập 8: Sử dụng hàm tính tổng

Thực hiện truy vấn để tính tổng số nhân viên trong bảng "NhanVien".

***Giải***

SELECT COUNT(\*)

FROM `quanlynhanvien`.`nhanvien`

AS tongsonhanvien

;

Bài tập 9: Sử dụng GROUP BY

Thực hiện truy vấn để tính tổng số nhân viên theo chức vụ trong bảng "NhanVien".

***Giải***

SELECT chuc\_vu, COUNT(\*)

FROM `quanlynhanvien`.`nhanvien`

AS tongsonhanvien

GROUP BY chuc\_vu

;

Bài tập 10: Truy vấn nâng cao với JOIN

Tạo một bảng mới có tên "PhongBan" chứa các trường: "id" (kiểu INT) và "ten\_phong" (kiểu VARCHAR). Thêm 3 bản ghi vào bảng "PhongBan" với thông tin phòng ban khác nhau.

***Giải***

USE quanlynhanvien;

CREATE TABLE PhongBan (

id INT,

ten\_phong VARCHAR(255)

)

SELECT \* FROM quanlynhanvien.phongban;

INSERT INTO phongban(id, ten\_phong)

VALUES

(1, 'phong ky thuat'),

(2, 'phong vat tu'),

(3, 'phong nhan su');

Bài tập 11: Truy vấn với INNER JOIN

Thực hiện truy vấn để lấy danh sách nhân viên cùng với tên phòng ban mà họ đang làm việc trong bảng "NhanVien" và "PhongBan".

Bài tập 12: Truy vấn với LEFT JOIN

Thực hiện truy vấn để lấy danh sách tất cả nhân viên cùng với tên phòng ban mà họ đang làm việc. Nếu nhân viên không có phòng ban thì hiển thị "Chưa xác định" trong bảng "NhanVien" và "PhongBan".

Bài tập 13: Truy vấn với WHERE và LIKE

Thực hiện truy vấn để tìm các nhân viên có chức vụ chứa từ khóa "kỹ" trong bảng "NhanVien".

Bài tập 14: Truy vấn với HAVING

Thực hiện truy vấn để tính tổng số nhân viên theo chức vụ trong bảng "NhanVien" và chỉ hiển thị các chức vụ có số lượng nhân viên lớn hơn 1.

Bài tập 15: Tạo VIEW

Tạo một view có tên "DanhSachNhanVien" để hiển thị thông tin về nhân viên từ bảng "NhanVien".