Thiết kế cơ sở dữ liệu

# Nguyên tắt thiết kế CSDL

* + Chính xác
  + Tránh trùng lặp
  + Dễ hiểu
  + Chọn đúng thuộc tính và kiểu thuộc tính
  + Chọn đúng mối quan hệ

# Các bước thiết kế CSDL



# Xác định các yêu cầu nghiệp vụ:

Đã có trong requirement.

Trả lời các câu hỏi:

* + CSDL sẽ được sử dụng như thế nào?
  + Những thông tin gì cần được lưu vào CSDL?

# Các bảng:

Note: Yellow là quan trọng cần làm trước

Green là làm sau khi đã làm xong các cái quan trọng

## Account (Tài khoản):

* id: khoá chính
* username: tên đăng nhập của người dùng
* email: địa chỉ email của người dùng
* password: mật khẩu của người dùng
* role: (chỉ chứa 4 giá trị : user, doctor, admin, hospital)
* id\_role:
* last\_login //nên để sau

## User (Thông tin khác của người dùng):

* id: Khoá chính
* name:
* \*\*birthday: ngày sinh của người dùng
* \*\*gender: giới tính của người dùng
* \*\*address: địa chỉ của người dùng
* \*\*phone: số điện thoại người dùng

## Doctor (Bác sĩ):

* id: Khoá chính
* name: họ và tên bác sĩ
* address: địa chỉ nơi làm việc bác sĩ
* id\_specialty: chuyên khoa của bác sĩ
* id\_hospital: bệnh viện mà bác sĩ đang làm việc
* phone: số điện thoại của bác sĩ
* birthday: ngày sinh của bác sĩ
* gender: giới tính của bác sĩ
* years\_of\_experience: số năm kinh nghiệm của bác sĩ
* descripe: mô tả
* price: giá một cuộc hẹn của bác sĩ
* average\_rating: điểm đánh giá trung bình//nên để sau

## Hospital (Bệnh viện – Hoặc phòm khám tư, dịch vụ):

* hospital\_id: Khoá chính
* name: tên bệnh viện
* email: email của bệnh viện
* address: địa chỉ bệnh viện
* ~~list\_id\_specialty~~: //cái này bỏ vì truy xuất trực tiếp từ bác sĩ ko cần phải lưu mảng dữ liệu (thường thi đối với mảng dữ liệu nếu có thể truy vấn từ bảng khác thì người ta sẽ ko lưu ngoài trừ trường hợp đặc biệt)
* ~~list\_id\_service~~: cung nhu tren

## Specialty (Chuyên khoa):

* id\_specialty: Khoá chính
* name: tên chuyên khoa
* descripe: mô tả chuyên khoa
* ~~total\_doctor:~~ tổng số bác sĩ //cái này bỏ vì đây à biến thay đổi liên tục cần truy vấn từ doctor ko nên gán trực tiếp

## Service (Dịch vụ):

* id\_service: Khoá chính
* name: tên dịch vụ
* descripe: mô tả dịch vụ

## Blog (Bài báo): //làm sau khi đã làm xong hết các chức năng chính

* Id:
* title: tiêu đề của bài báo
* id\_ author: tác giả của bài báo (khoá ngoại đến ‘bác sĩ’)
* content: nội dung của bài báo
* image: hình ảnh đại diện cho bài báo
* id\_category: chuyên mục của bài báo
* views: số lượt xem bài viết

## Category (chuyên mục sức khoẻ): //làm khi đã xong các chức năng chính

* category\_id: Khoá chính
* name: Tên chuyên mục
* describe: Mô tả của chuyên mục đó

## Schedule (lịch làm việc của bác sĩ):

* id: khoá chính
* id\_doctor: khoá ngoại liên kết đến bác sĩ
* day\_of\_week: ngày trong tuần (2 ~ thứ 2 ,3 ~ thứ 3, …, 8 ~ cn)
* time\_start: giờ bắt đầu làm việc
* time\_end: giờ kết thúc làm việc

## Appointment (Lịch hẹn):

* id: Khoá chính
* id\_user: khoá ngoại liên kết đến User
* id\_schedule: khoá ngoại liên kết đến Schedule
* date: ngày hẹn
* time: giờ hẹn
* rating: Điểm đánh giá từ người dùng.
* comment: Bình luận hoặc đánh giá chi tiết từ người dùng.

## Calculator (Công cụ tính toán)

* id: Khóa chính để xác định mỗi công cụ tính toán.
* name: Tên công cụ tính toán, ví dụ: "BMI Calculator".
* description: Mô tả công cụ tính toán.
* formula: Công thức tính toán của công cụ. VD: “(a+b)\*c”
* questions: Mảng chứa các câu hỏi cho từng tham số. VD: [“Bạn bao nhiêu tuổi? (năm)”, “Bạn cao bao nhiêu? (cm)”, “Cân nặng của bạn (kg)”]

## Save (Bảng lưu các blog của người dùng):

* id: Khoá chính
* id\_user: người dùng
* id\_blog: blog mà người dùng muốn lưu
* time: Thời gian lưu

## Bảng dịch vụ - bác sĩ

* id: Khoá chính
* id\_service: dịch vụ
* id\_doctor: bác sĩ

# Quan trọng:

## Cách nạp công cụ tính toán vào trong cơ sở dữ liệu:

Thông qua hàm trên chứa các tham số có thể lưu trữ trong cơ sở dữ liệu:

str: String

array: Array

function abc(str, array) {

let expression = str;

// Thay thế các ký tự trong biểu thức bằng các giá trị tương ứng từ mảng

for (let i = 0; i < array.length; i++) {

const variable = String.fromCharCode(97 + i); // Chuyển đổi thành ký tự a, b, c, ...

expression = expression.replace(new RegExp(variable, 'g'), array[i]);

}

// Tính toán biểu thức

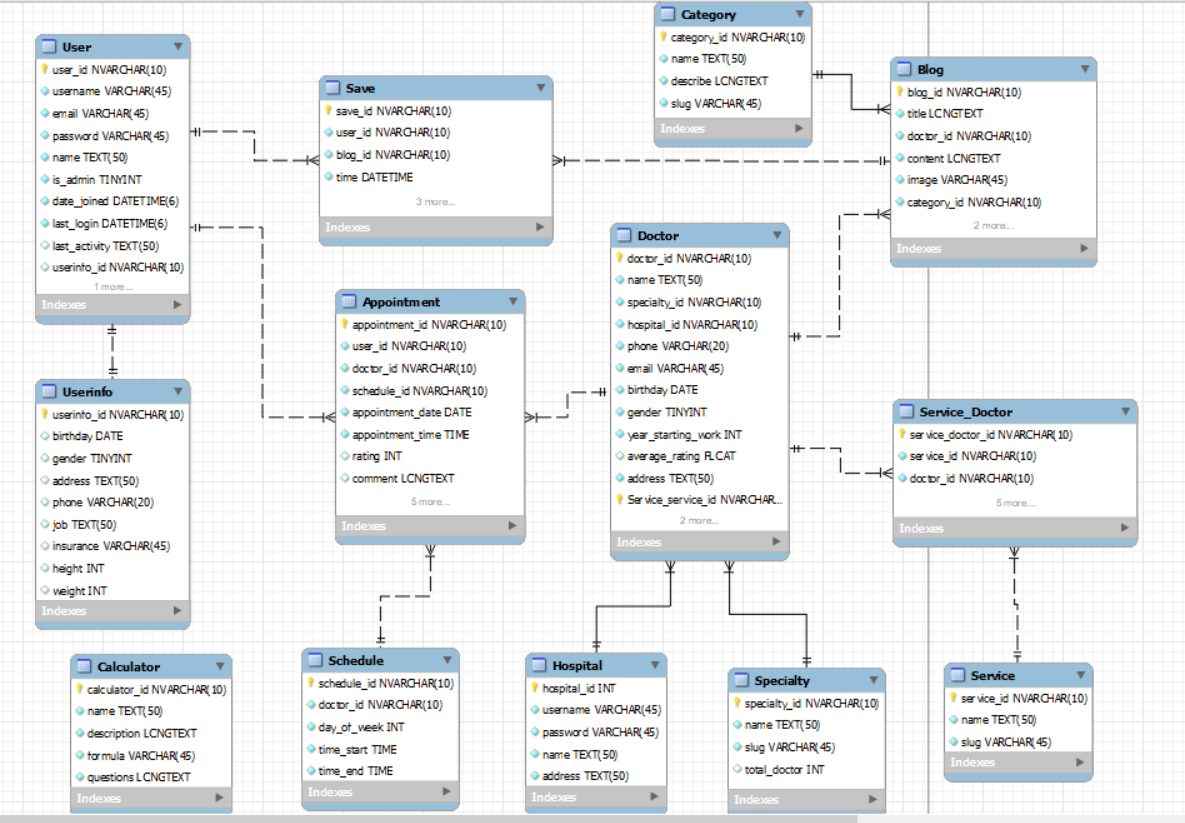
return eval(expression); // Lưu ý: Sử dụng eval() cần cẩn thận và đảm bảo an toàn dữ liệu đầu vào.

}

// Sử dụng hàm abc

const result = abc("(a+b)\*c", [2, 3, 5]);

console.log(result); // Kết quả: 30



Mỗi lần gọi API của blog thì mình sẽ thêm trường lượt views lên.