# Excercise

(\*): Cách làm bài: Dưới mỗi câu hỏi hãy chụp ảnh screenshot bao gồm câu SQL và kết quả.

(\*) Để kiểm tra xem trường có bị NULL hay không thì dùng IS NULL. Ví dụ

SELECT \* FROM … WHERE song\_name IS NULL.

## Sử dụng cơ sở dữ liệu sau

Sử dụng cơ sở dữ liệu mẫu trong thư mục Materials bao gồm

* schema.sql: Định nghĩa schema
* data.sql: Chưa data để import vào CSDL

Thực hiện các bài tập sau

* Nhập dữ liệu từ bảng **schema.sql** trên vào cơ sở dữ liệu.
* Tạo thêm một bảng tên là “**Songs**” bao gồm các trường như sau
  + id: Primary key, Auto-increment, unsigned
  + name: String có chiều dài 255 (không được null)
  + length: float, chỉ định thời gian bài hát tính theo phút (không được null)
  + album\_id: integer, là khoá ngoại và trỏ đến bảng album với khoá chính id
* Nhập dữ liệu trong file **data.sql**
* Viết câu sql query để lấy trường **name** của bảng **bands**. Hãy thay đổi tên cột kết quả trả về là **Band Name** thay vì **name,** sử dụng ‘as’ trong SQL.

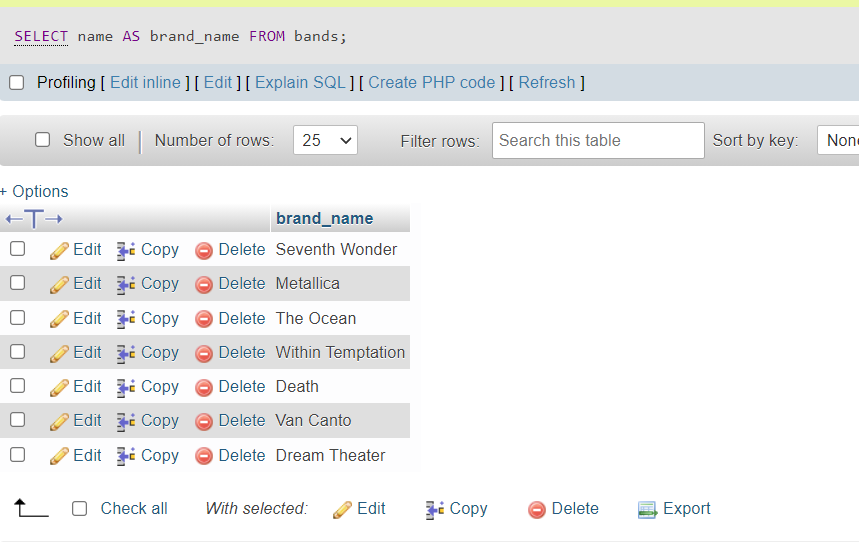


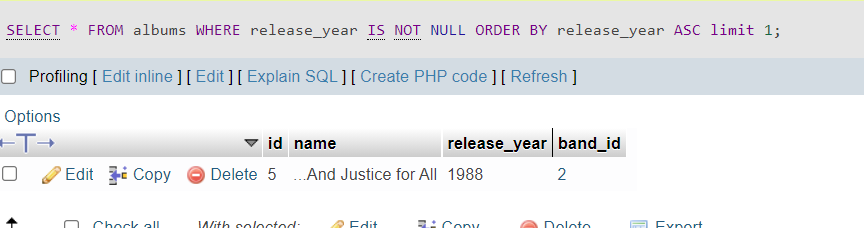


Figure 1: Ví dụ kết quả trả về

* Viết câu sql xác định thông tin của album được phát hành sớm nhất (chỉ trả về 1 kết quả, không tính giá trị NULL)



Figure 2: Album được phát hành sớm nhất



* Lấy ra danh sách các bands mà có thông tin album ứng với bands đó, kết quả trả về sẽ như sau



Figure 3: Thông tin của các bands

Gợi ý: Sử dụng DISTINCT(column) để lấy giá trị duy nhất của bảng. VD nếu như Seventh Wonder trong cột Band Name xuất hiện nhiều hơn 1 lần thì chỉ lấy ra 1.



* Ngược lại với câu trên, hãy lấy ra danh sách các bands không có album đi kèm

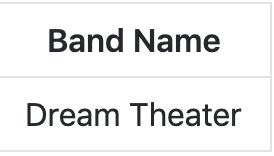
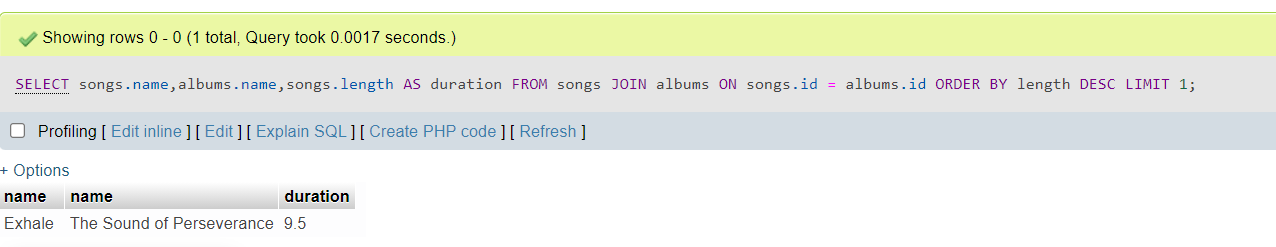


Figure 4: Các bands không có album đi kèm

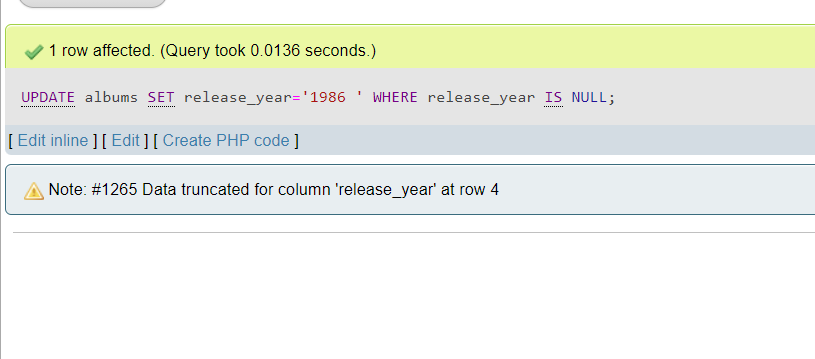
* Lấy album có tổng độ dài là lâu nhất. Kết quả trả về bao gồm tên Bands sở hữu album đó, năm phát hành album và tổng thời gian của album

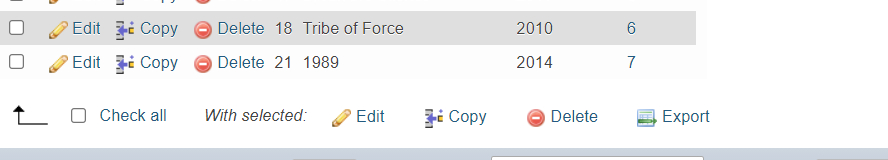


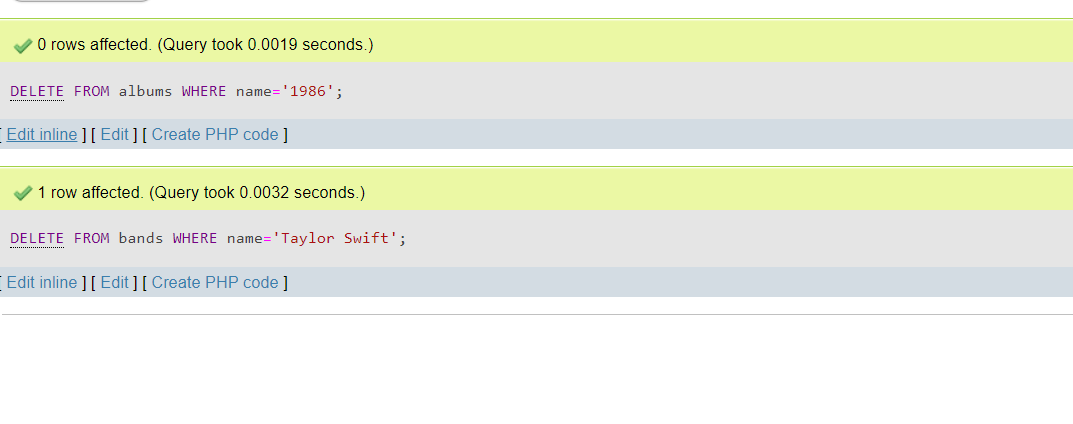
Figure 5: Kết quả



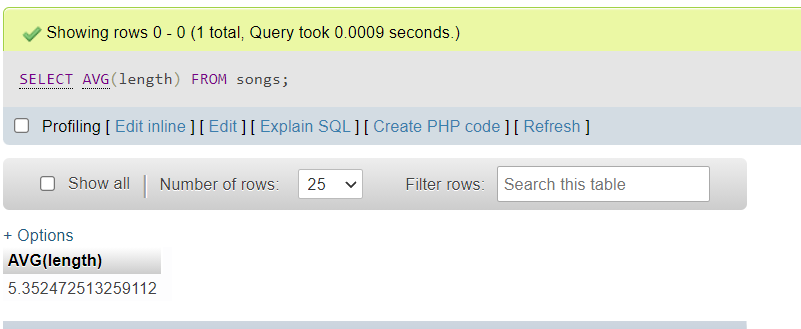
* Cập nhật thông tin album, đặt năm phát hành của album là 1986 với trường hợp release\_year của album bị NULL trước đó



* Thêm vào cơ sở dữ liệu 1 record của bands và một record của album ứng với bands đó mà bạn thích.
* 
* Xoá thông tin của bands và album vừa tạo



* Lấy thông tin tổng thời gian trung bình của album



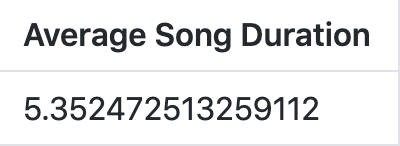
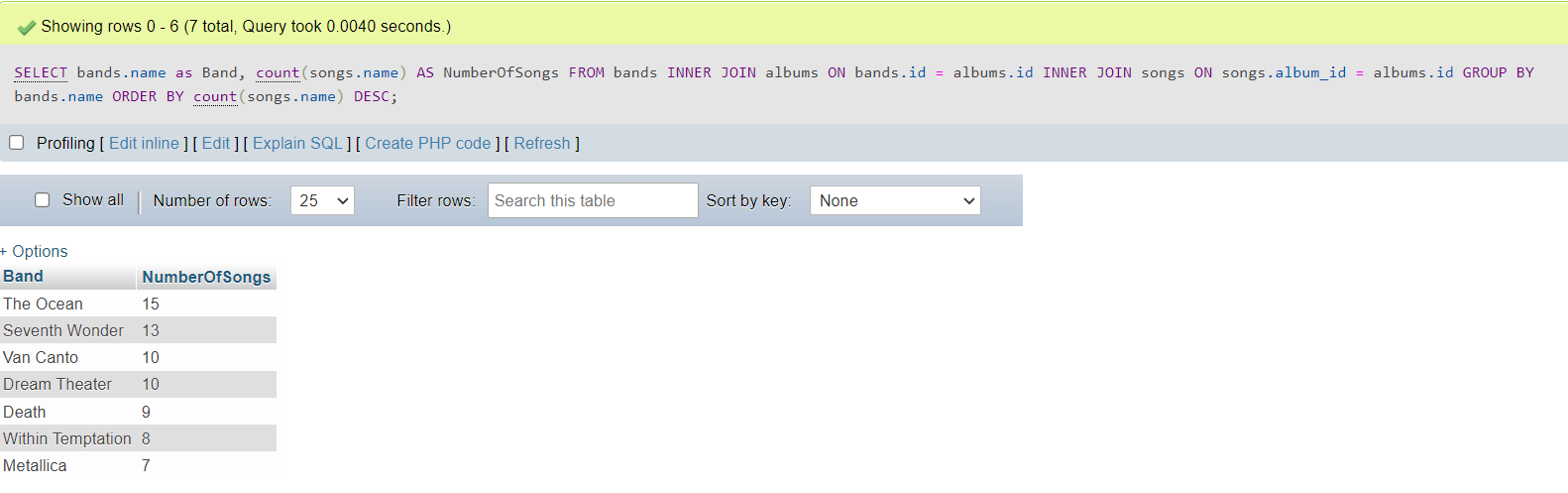


Figure 6: Thời gian trung bình của album

* Lấy số lượng bài hát của từng band



SELECT bands.name as Band, count(songs.name) AS NumberOfSongs

FROM bands

INNER JOIN albums ON bands.id = albums.id

INNER JOIN songs ON songs.album\_id = albums.id

GROUP BY bands.name

ORDER BY count(songs.name) DESC

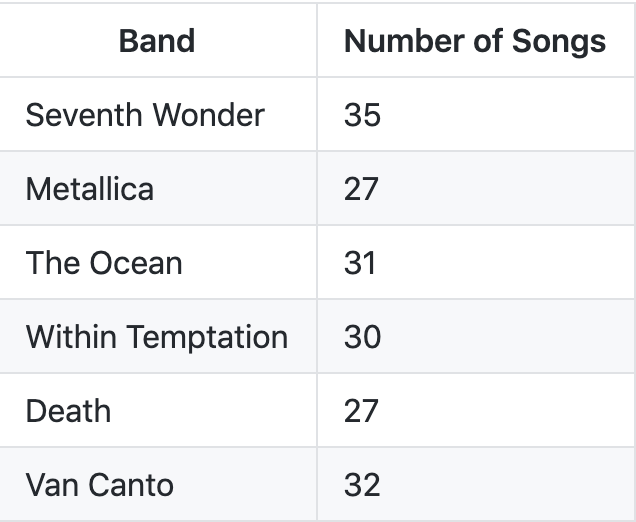


Figure 8: Số lượng bài hát của bands

## Sử dụng model sau để xây dựng cơ sở dữ liệu như sau và viết SQL để thoả mãn điều kiện dưới đây

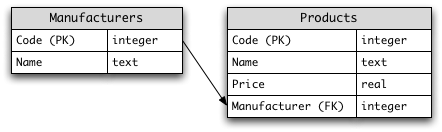


Figure 9: ER model

* Xây dựng cơ sở dữ liệu như ở hình trên
* Sau khi xây dựng xong CSDL thì tiến hành import data vào cơ sở dữ liệu vừa tạo.
* Lấy danh sách tên của tất cả products
* Lấy danh sách tên và giá của tất cả products
* Lấy danh sách products mà price nhỏ hơn 200
* Lấy danh sách products mà price nằm trong khoảng 60 đến 200
* Tính giá trung bình cho toàn bộ products
* Tính giá trung bình cho toàn bộ products mà có manufacturer = 2
* Tính xem có bao nhiêu sản phẩm với giá cả >= 180, sắp xếp theo price giảm dần, sau đó đến tên theo chiều tăng dần
* Viết câu query để lấy sản phẩm với giá rẻ nhất.
* Viết câu query để thêm một sản phẩm mới với thông tin như sau
  + Tên: Loudspeaker
  + Price: 70
  + Manufacturer: 2
* Cập nhật thông tin của sản phẩm với product\_id = 8 với tên là “Laser Printer”
* Cập nhật thông tin cho TẤT CẢ sản phẩm, đặt giá mới bằng 90% giá cũ

## Sử dụng model sau để xây dựng cơ sở dữ liệu như sau và viết SQL để thoả mãn điều kiện dưới đây

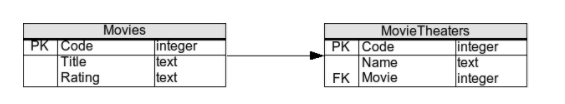


Figure 10: CSDL

* Xây dựng cơ sở dữ liệu như ở hình trên
* Sau khi xây dựng xong CSDL thì tiến hành import data vào cơ sở dữ liệu vừa tạo (bao gồm 2 files: Movies.sql và MoviesTheaters.sql)
* Lấy ra title của tất cả movies
* Lấy ra danh sách ratings khác nhau (không được trùng lặp) của movies
* Lấy ra danh sách các bộ phim không có ratings
* Lấy ra danh sách các rạp chiếu phim không có phim
* Lấy ra danh sách các theaters và các bộ phim ứng với theaters đó (kể cả các theaters không có bộ phim nào cũng cần phải trả về kết quả).
* Lấy danh sách các bộ phim, và các theaters ứng với bộ phim đó (kể cả bộ phim đó không có theaters nào vẫn cần phải trả ra kết quả)
* Lấy title của tất cả các bộ phim không được trình chiếu ở bất kì theaters nào
* Thêm mới một record với thông tin như sau
  + Title: ‘One, Two, Three’
  + Rating: NULL
* Cập nhật thông tin của tất cả các bộ phim không có rating thành “G”
* Xoá tất cả các theaters mà phim của theaters đó có Rating là ‘NC-17’

## Bài tập tổng hợp từ đầu

Sử dụng Bootstrap và PHP để hoàn thành bài tập sau đây

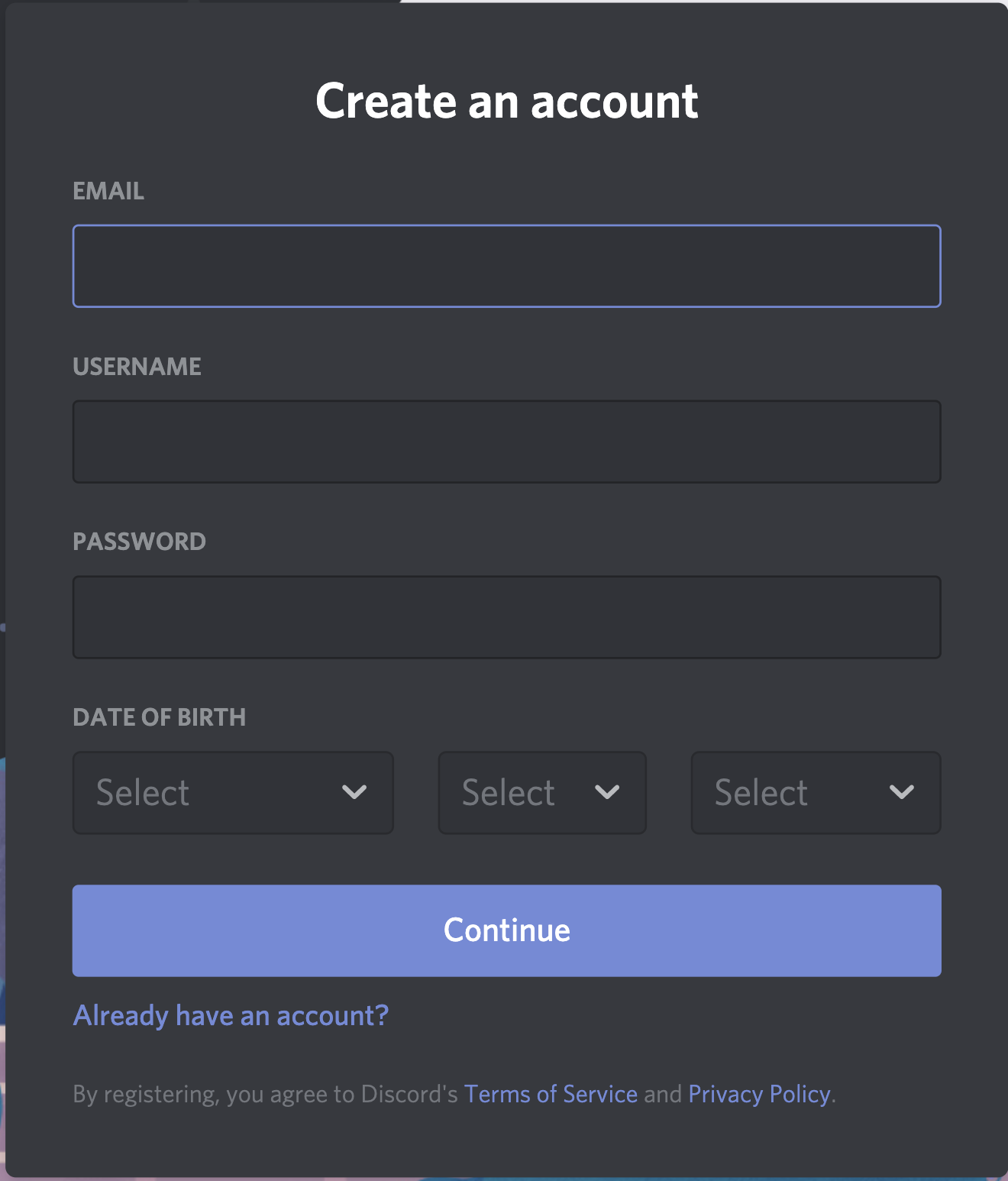


Figure : Giao diện đăng kí của trang Discord

Sử dụng Bootstrap và HTML để xây dựng giao diện đăng kí và thực hiện gửi giá trị của Form lên server. Cần validate các trường như sau:

* Email: Bắt buộc nhập và phải bắt buộc có định dạng email
* Username: Bắt buộc nhập
* Password: Bắt buộc nhập
* Date of birth: Day month và year bắt buộc phải nhập

Cho biết màu của background color là #36393f, đồng thời, màu của nút Continue là #7289da.

Khi không có lỗi liên quan đến validation thì tiến hành gửi form lên server. Ghi rõ thông tin đăng kí của người dùng bên dưới form.

Đồng thơi sau khi đăng kí thành công, sử dụng Session để lưu thông tin người dùng và khi chuyển sang một trang khác thì ghi rõ: **Welcome $username.** Ví dụ **Welcome camnh**

## Bài tập 4.1: Bổ sung form đăng kí ở trên

Thêm một trường để upload ảnh đại diện avatar lên. Ảnh chỉ được phép có phần mở rộng là jpg và png. Đồng thời dung lượng tối đa khi upload ảnh lên là 2MB. Trong trường hợp ảnh vượt quá 2MB và đuôi mở rộng không phải là **png** hoặc **jpg** thì thông báo lỗi.

Hiển thị ảnh cùng với các thông tin ở trên bài 4.1 ra ngoài nếu không gặp lỗi gì.