

Cấu trúc hệ thống Views

1. Page Artists:

- Tổng số nghệ sĩ **[1.1]**
- Một word cloud thể hiện mức độ nổi bật của các nghệ sĩ dựa trên các yếu tố khác nhau. (sắp xếp popularity từ trên xuống) **[1.2]** (**Không thể sử dụng câu lệnh ORDER BY trong khi tạo View nên khi query từ view này vui lòng thêm syntax: ORDER BY population**)
- Một bảng hiển thị các nghệ sĩ được xếp hạng theo số lượng người theo dõi.(sắp xếp theo số lượng followers) **[1.3]** (**Không thể sử dụng câu lệnh ORDER BY trong khi tạo View nên khi query từ view này vui lòng thêm syntax: ORDER BY followers**)
- Biểu đồ scatter thể hiện mối quan hệ giữa số lượng người theo dõi và phổ biến của nghệ sĩ. (Chỉ lấy các field followers và popularity của các nghệ sĩ theo name) **[1.4]**
- View phục vụ cho hệ thống Store Procedures Filter nên những mối quan hệ giữa Artist và Track (sẽ có gọi từ View [3.1] để lấy thêm cảm thông tin của những track đó) **[1.7]**:
 - + SP Filter theo tên nghệ sĩ : Param nhập vào sẽ là tên nghệ sĩ. **[1.7.1]** (Query bằng câu lệnh : `EXEC [pds].[SP_1_7_1] @ArtistName = ''`)
- View phục vụ cho hệ thống Store Procedures Filter nên những mối quan hệ giữa Artist và Geners **[1.6]**:
 - + SP Filter theo tên nghệ sĩ : Param nhập vào sẽ là tên nghệ sĩ. **[1.6.1]** (Query bằng câu lệnh : `EXEC [pds].[SP_1_6_1] @ArtistName = ''`)
- View phục vụ cho hệ thống Store Procedures Filter nên những mối quan hệ giữa Artist và Album **[1.5]**:
 - + SP Filter theo tên nghệ sĩ : Param nhập vào sẽ là tên nghệ sĩ. **[1.5.1]** (Query bằng câu lệnh : `EXEC [pds].[SP_1_5_1] @ArtistName = ''`)

2. Page Albums:

- View phục vụ cho hệ thống Store Procedures Filter nên những mối quan hệ giữa Albums và Track (sẽ có gọi từ View [3.1] để lấy thêm thông tin của feature từ những track đó) **[2.1]**

3. Page Tracks:

- View phục vụ cho hệ thống lấy hết những feature từ *track_feature* sang cho *track* **[3.1]**

Thứ tự tạo View: 1.1 -> 1.6 trước, sau đó tạo [3.1] -> sau đó tạo [1.7] sau đó tạo [2.1]