

MÔ PHỎNG HỆ THỐNG QUẢN LÝ VÀ PHÁT HÀNH NHẠC SÓ SPOTIFY



TRẦN XUÂN BẢO, PHAN HOÀNG DŨNG, HÀ XUÂN HUY

Institute of Artificial Intelligence, University of Engineer and Technology

Nội dung

1

GIỚI THIỆU, TỔNG QUAN VỀ DATABASE

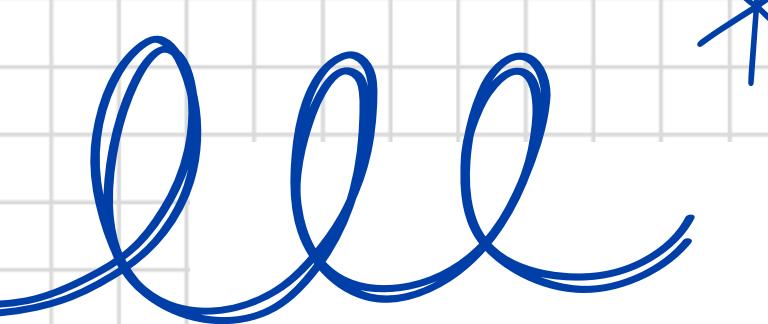
2

QUÁ TRÌNH THỰC HIỆN

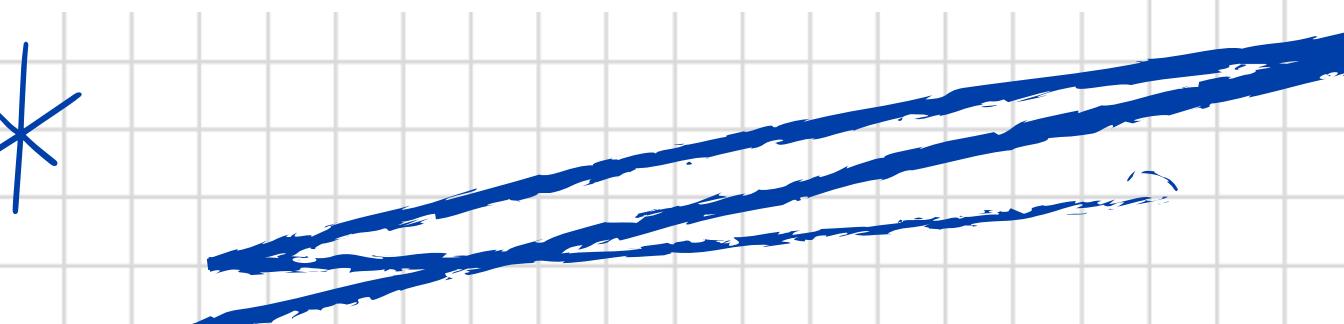
3

HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI





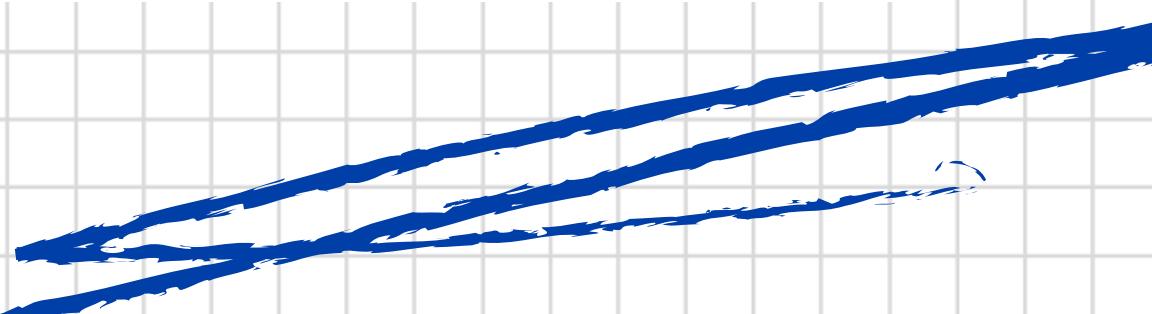
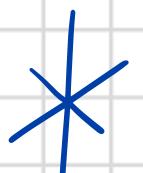
1. Giới thiệu



- Spotify - Nền tảng phát nhạc hàng đầu thế giới:
 - Hơn 500 triệu người dùng mỗi tháng, xử lý hàng tỷ bản ghi liên quan đến bài hát, nghệ sĩ và người dùng.
 - Là hệ thống cơ sở dữ liệu lớn và phức tạp, yêu cầu thiết kế tối ưu và hiệu suất cao.
- Dự án này được thực hiện nhằm mô phỏng hệ thống cơ sở dữ liệu của Spotify, giúp sinh viên:
 - Áp dụng kiến thức thiết kế và tối ưu cơ sở dữ liệu vào thực tế.
 - Hiểu cách quản lý và phân tích dữ liệu quy mô lớn trong hệ thống âm nhạc hiện đại.

Giới thiệu

- Tại sao chọn Spotify?
 - Là hình mẫu điển hình cho cơ sở dữ liệu lớn, hiện đại, có ứng dụng thực tiễn cao.
 - Thành công của Spotify phụ thuộc vào hệ thống cơ sở dữ liệu mạnh mẽ, xử lý hiệu quả lượng dữ liệu khổng lồ.
- Kết quả dự án là một hệ thống cơ sở dữ liệu MySQL hoàn chỉnh, đáp ứng các tính năng cốt lõi như quản lý người dùng, bài hát, playlist và gợi ý cá nhân hóa, đồng thời đảm bảo hiệu năng và khả năng mở rộng.



Tổng quan về database

Các tính năng chính:

- Quản lý người dùng: Lưu thông tin hồ sơ và sở thích người dùng.
- Quản lý bài hát: Tổ chức dữ liệu bài hát, nghệ sĩ, album.
- Quản lý playlist: Hỗ trợ tạo playlist cá nhân hóa.
- Đề xuất cá nhân hóa: Mô phỏng gợi ý bài hát dựa trên sở thích người dùng.

Công cụ sử dụng:

- Hệ quản trị cơ sở dữ liệu: MySQL
- Công cụ mô hình: ERD tools (Draw.io)
- Môi trường phát triển: SQL và công cụ dòng lệnh



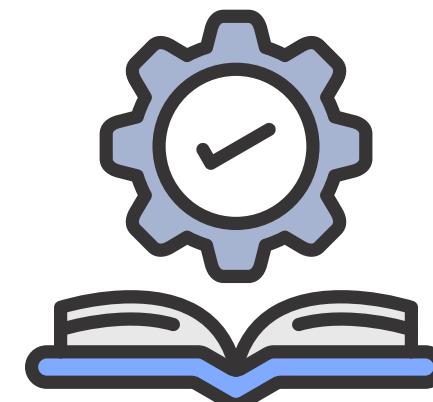
Tổng quan về database

Kết quả:

- Mô hình dữ liệu: Sơ đồ ERD chi tiết.
- Triển khai cơ sở dữ liệu: Hệ thống MySQL với đầy đủ bảng và quan hệ.
- Truy vấn SQL:
 - Đăng ký và cập nhật hồ sơ người dùng.
 - Quản lý bài hát và playlist.
 - Gợi ý bài hát cá nhân hóa.
- Tối ưu hiệu năng: Đảm bảo xử lý lượng lớn dữ liệu nhanh và hiệu quả.

Bài học kinh nghiệm:

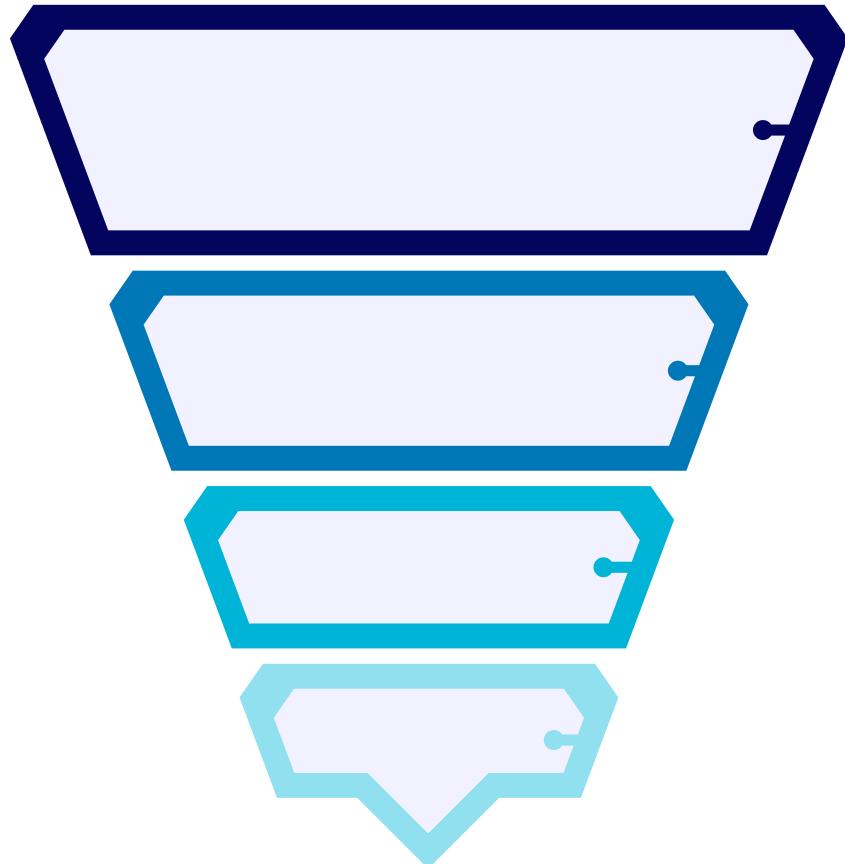
- Thiết kế cơ sở dữ liệu có cấu trúc khoa học cho hệ thống lớn.
- Thực hiện truy vấn SQL phức tạp cho bài toán thực tế.
- Tối ưu cơ sở dữ liệu để đạt hiệu suất cao.
- Kết nối lý thuyết với thực hành trong thiết kế cơ sở dữ liệu.



2. Quá trình thực hiện

Dự án được triển khai qua các bước từ phân tích yêu cầu đến xây dựng hệ thống cơ sở dữ liệu hoàn chỉnh. Mỗi giai đoạn tập trung vào việc tái hiện chính xác trải nghiệm người dùng và đảm bảo hiệu năng cao, bao gồm:

- Phân tích và thiết kế mô hình dữ liệu.
- Triển khai cơ sở dữ liệu với các bảng và quan hệ trong MySQL.
- Xây dựng và kiểm thử các truy vấn SQL phục vụ chức năng cốt lõi.
- Tối ưu hệ thống để đáp ứng yêu cầu xử lý dữ liệu lớn và đảm bảo khả năng mở rộng.

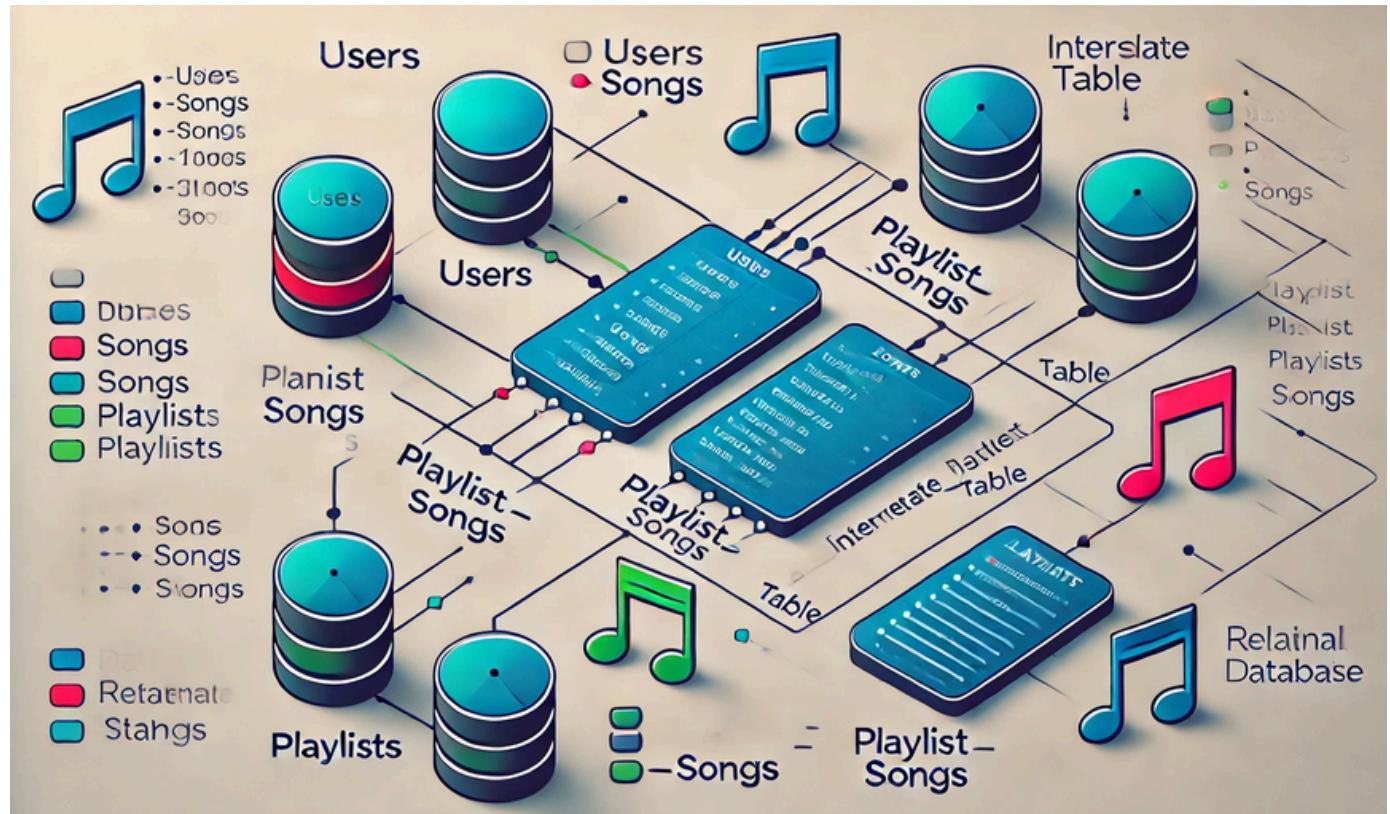


Thành phần của database

Các thực thể chính:

- Người dùng (Users): Đại diện cho người nghe. Bao gồm **user_id, name, email, password, date_joined**.
- Bài hát (Songs): Đại diện cho các bài hát. Bao gồm **song_id, title, album_id, duration, genre**.
- Nghệ sĩ (Artists): Đại diện cho những người sáng tác bài hát và album. Bao gồm **artist_id, name**.
- Bộ sưu tập (Albums): Đại diện cho bộ sưu tập các bài hát. Bao gồm **album_id, album_name**.
- Thể loại (Genre): Đại gồm diện cho các thể loại âm nhạc. Bao gồm **genre_id, name**.
- Gợi ý (Recommendation): Đại diện cho các gợi ý hệ thống đưa ra. Bao gồm **recommendation_id, user_id, song_id, recommendation_reason, time_stamp**.
- Lịch sử nghe (Playback history): Đại diện lượt nghe của người dùng. Bao gồm **user_id, song_id, total_time**.
- Danh sách phát (Playlists): Đại diện danh sách bài hát. Bao gồm **playlist_id, playlist_name, user_id**.
- Gói đăng ký (Subscriptions): Theo dõi các gói đăng ký của người dùng. Bao gồm **subscription_id, user_id, subscription_type, start_date, end_date**.

Thành phần của database



Các bảng trung gian:

- SongContributor (Many-to-Many giữa Song và Artist):
SongContributor(song_id (FK), artist_id (FK))
- SongCategories (Many-to-Many giữa Song và Genre):
SongCategories(song_id (FK), genre_id (FK))
- AlbumCategories (Many-to-Many giữa Album và Genre):
AlbumCategories(album_id (FK), genre_id (FK))
- ArtistStyles (Many-to-Many giữa Artist và Genre):
ArtistStyles(artist_id (FK), genre_id (FK))

* Quy trình hoạt động và tính năng chính

- Người dùng:
 - Đăng nhập bằng email và mật khẩu.
 - Tìm kiếm bài hát, nghệ sĩ, album theo từ khóa.
 - Quản lý playlist cá nhân: Tạo, thêm, xóa bài hát dễ dàng.
 - Gợi ý cá nhân hóa: Khám phá nội dung mới dựa trên thói quen nghe nhạc và sở thích.
 - Thanh toán Premium: Quản lý gói đăng ký và thông báo khi sắp hết hạn.
- Nghệ sĩ:
 - Đăng nhập và quản lý nội dung cá nhân (bài hát, album, thể loại).
 - Thống kê lượt nghe: Theo dõi mức độ tương tác của người dùng.
 - Hỗ trợ phát triển sự nghiệp âm nhạc thông qua tương tác trực tiếp với khán giả.



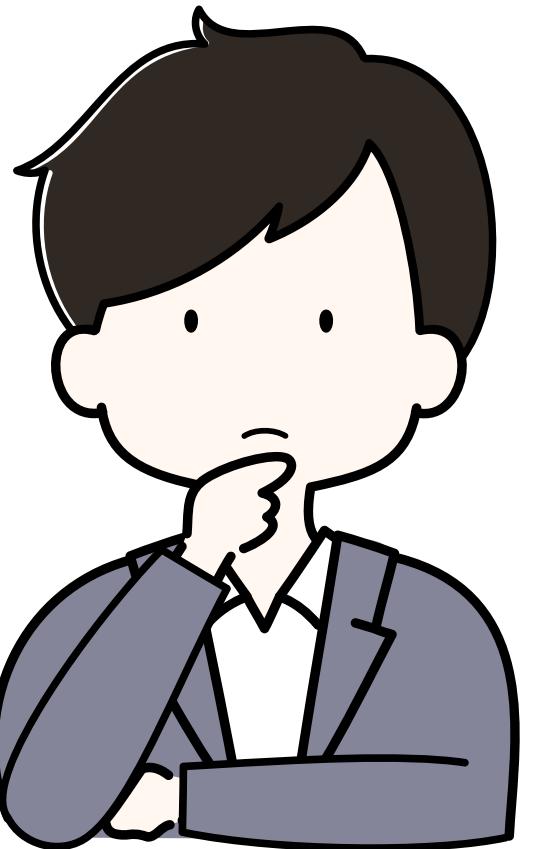


Một số vấn đề gặp phải



1. Lựa chọn các thực thể

- **Vấn đề:**
 - Genre nên là một thuộc tính hay một thực thể?
 - Các thông tin chi tiết khác như mô tả, nguồn gốc cần lưu trữ ở đâu?
- **Giải pháp:** Chọn Genre làm một thực thể riêng để bổ sung thêm các thông tin chi tiết.
- **Độ hiệu quả:**
 - **Ưu điểm:** Dễ mở rộng thông tin về genre mà không cần thay đổi cấu trúc bảng khác.
 - **Nhược điểm:** Phát sinh thêm quan hệ và truy vấn phức tạp hơn.



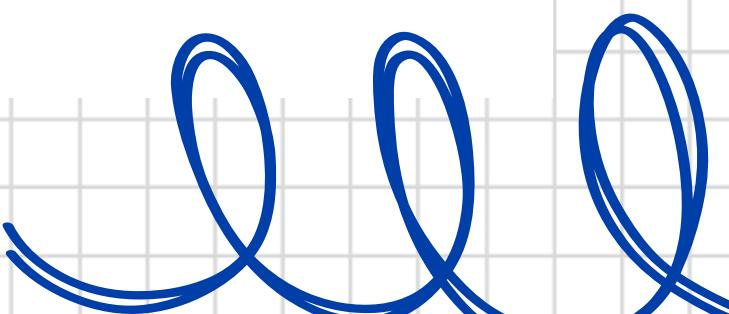
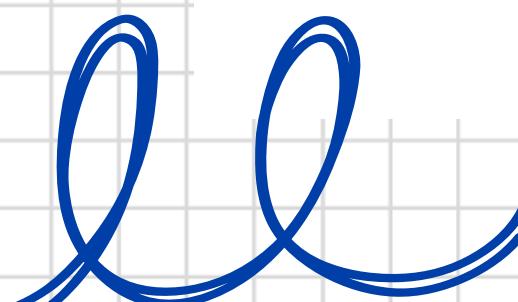
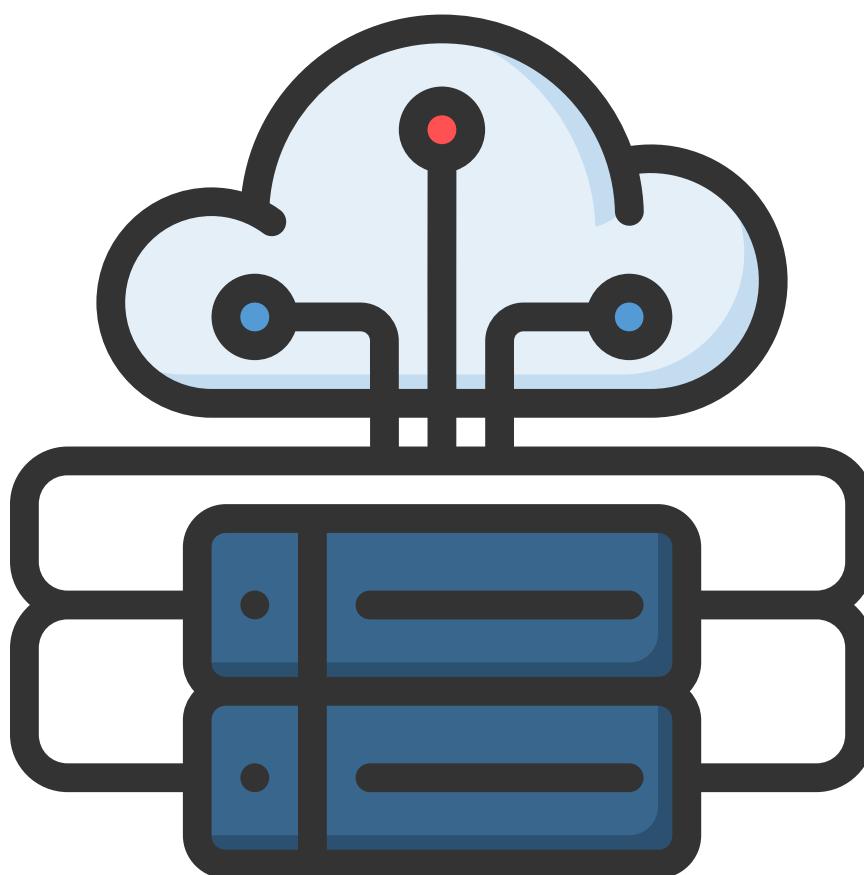


Một số vấn đề gặp phải



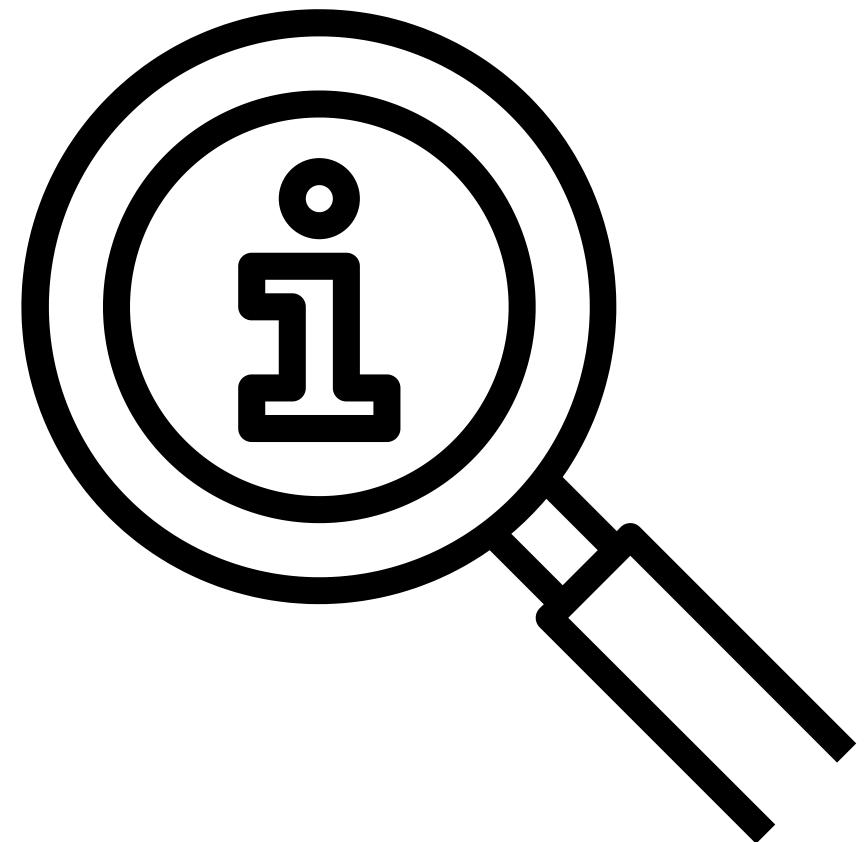
2. Lưu trữ PlaybackHistory

- Vấn đề: Nên lưu tổng thời gian nghe hay từng lần nghe chi tiết?
- Giải pháp: Chọn lưu tổng thời gian nghe để giảm dung lượng lưu trữ.
- Độ hiệu quả:
 - Ưu điểm: Tiết kiệm dung lượng lưu trữ.
 - Nhược điểm: Mất thông tin chi tiết của từng lượt nghe.



*

Một số vấn đề gấp phải



3. Lưu trữ thông tin nghệ sĩ và album

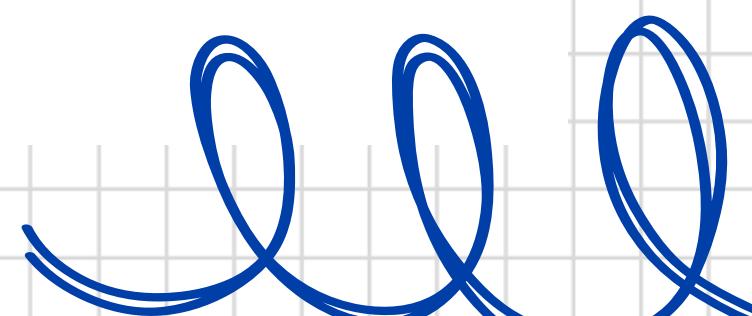
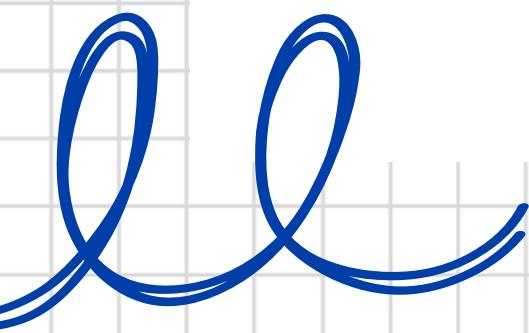
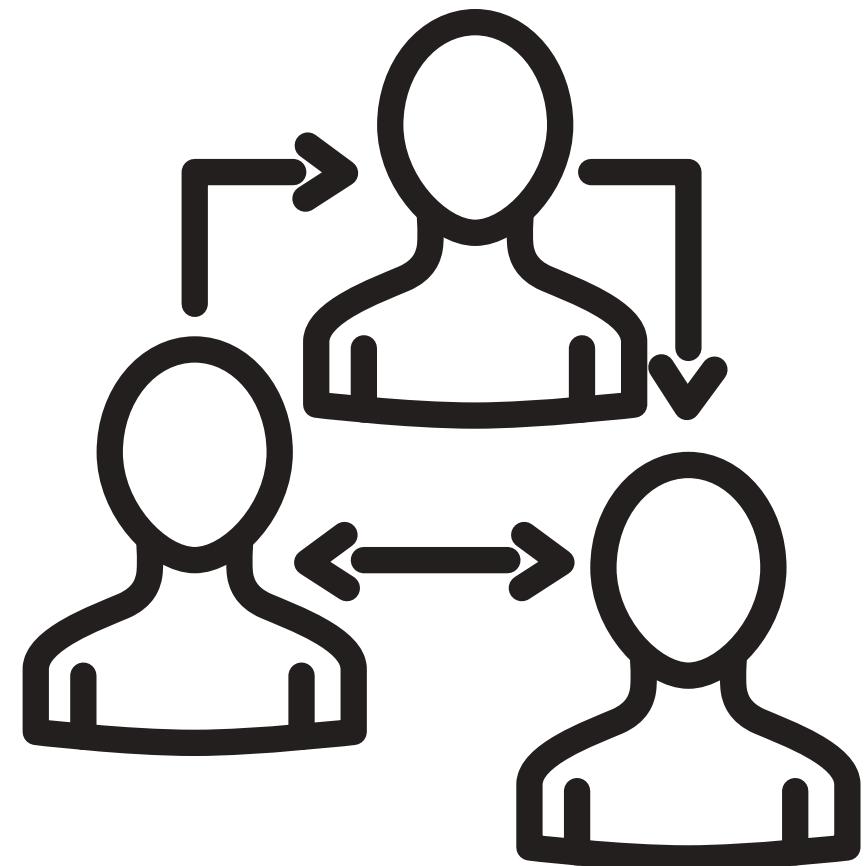
- Vấn đề: Quốc tịch nghệ sĩ, ngày phát hành album cần lưu không?
- Giải pháp: Tạm thời bỏ qua do không đủ dữ liệu đầu vào.
- Độ hiệu quả: Tiết kiệm thời gian triển khai nhưng làm giảm chi tiết hệ thống.

Một số vấn đề gặp phải



4. Quan hệ giữa các thực thể (User và Subscription)

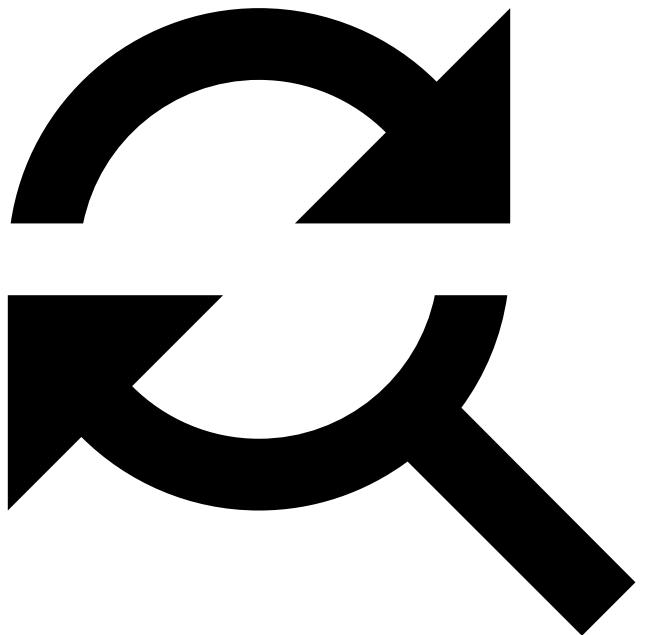
- Vấn đề: Cách quản lý gói đăng ký, kể cả người dùng miễn phí.
- Giải pháp: Đặt mặc định gói của người dùng miễn phí là "Free".
- Độ hiệu quả:
 - Ưu điểm: Đơn giản hóa việc quản lý và truy vấn gói đăng ký.
 - Nhược điểm: Không thể ghi nhận người dùng không có gói đăng ký (nếu cần).



Một số lệnh SQL

1. Tự động ghi nhận ngày tham gia khi thêm người dùng

- Yêu cầu: Tự động cập nhật ngày tham gia khi thêm mới người dùng.
- Nhóm sử dụng trong trường hợp:
 - Khi thêm mới người dùng vào hệ thống mà không cần nhập thủ công ngày tham gia.
 - Đảm bảo rằng tất cả các bản ghi trong bảng User đều có ngày tham gia hợp lệ, không bị bỏ trống (NULL).
 - Giúp phát hiện lỗi nhập liệu khi thiếu thông tin ngày tham gia.



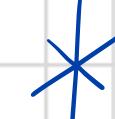
Một số lệnh SQL

Với trigger này, nhóm đảm bảo rằng mỗi khi một người dùng mới được thêm vào bảng `Users`, trường `date_joined` sẽ được tự động điền ngày hiện tại, giúp duy trì tính nhất quán và chính xác trong cơ sở dữ liệu mà không cần can thiệp thủ công.

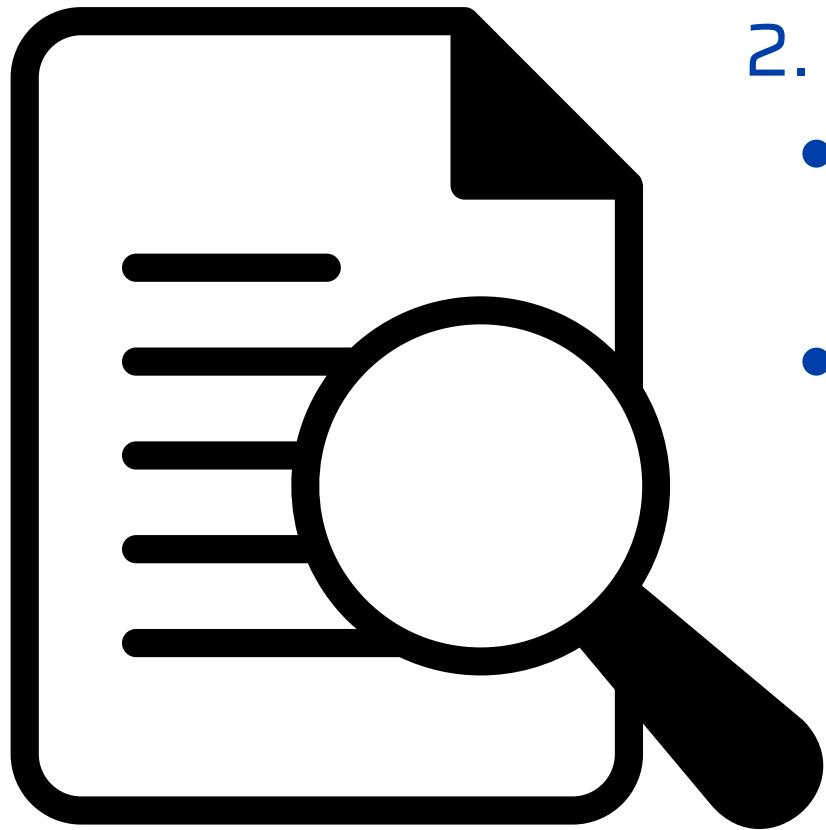
```
-- TRIGGER: automatically set date_joined to the current date when a new user is added
DELIMITER $$

CREATE TRIGGER set_date_joined
BEFORE INSERT ON Users
FOR EACH ROW
BEGIN

    IF NEW.date_joined IS NULL THEN
        SET NEW.date_joined = CURDATE();
    END IF;
END$$
```



Một số lệnh SQL



2. Tìm kiếm lịch sử nghe nhạc của người dùng từ PlaybackHistory
- Yêu cầu: Truy vấn lịch sử nghe nhạc của người dùng, bao gồm tên người dùng và tiêu đề bài hát.
 - Nhóm sử dụng trong trường hợp:
 - Khi người dùng muốn xem lại danh sách các bài hát họ đã nghe.
 - Đội hỗ trợ khách hàng cần tra cứu thông tin lịch sử nghe nhạc của một người dùng cụ thể để xử lý khiếu nại.
 - Hệ thống phân tích cần lấy dữ liệu để đánh giá hành vi nghe nhạc của từng người dùng.



Một số lệnh SQL

Câu truy vấn sử dụng INNER JOIN để lấy tên người dùng và các bài hát họ đã nghe. Chỉ trả về kết quả khi có bản ghi khớp giữa các bảng Users, PlaybackHistory, và Songs, đảm bảo mỗi người dùng và bài hát đều có liên kết hợp lệ.

```
-- INNER JOIN: retrieves the names of users and the songs they have played
SELECT name AS user_name, Songs.title AS song_title
FROM Users
INNER JOIN PlaybackHistory ON Users.user_id = PlaybackHistory.user_id
INNER JOIN Songs ON PlaybackHistory.song_id = Songs.song_id;
```

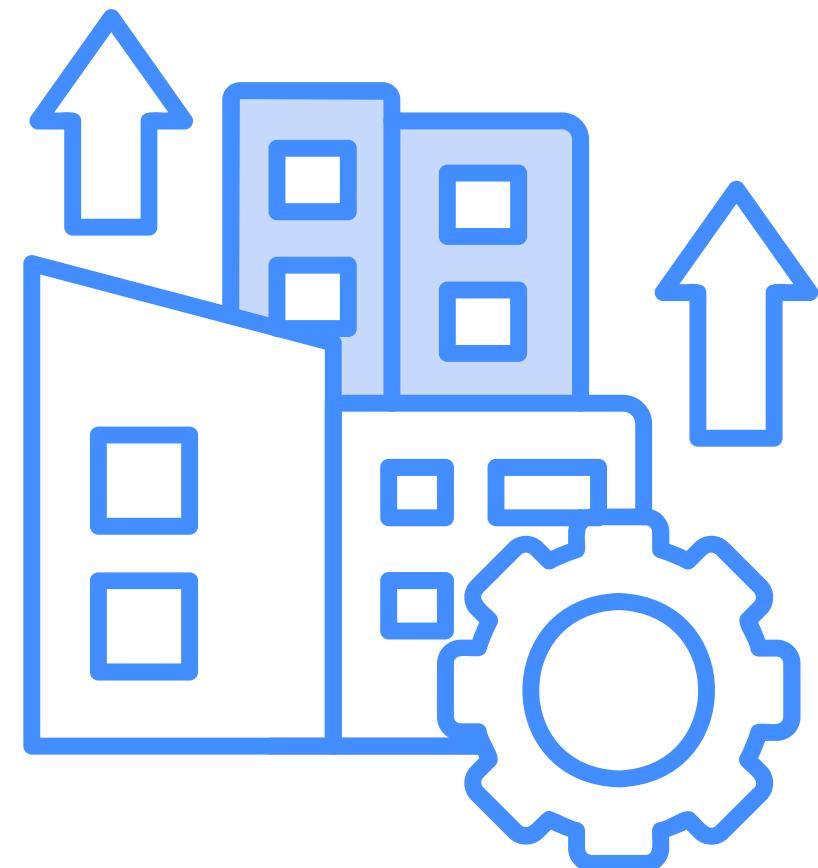
Kết quả

	user_name	song_title
▶	Alice Johnson	Vermelhos de Ciclos
	Alice Johnson	Sincronia
	Alice Johnson	Apricots
	Alice Johnson	Lido
	Alice Johnson	Sentient
	Bob Williams	Atlas
	Bob Williams	Sleigh Ride
	Bob Williams	Rudolph The Red-Nosed Reindeer
	Bob Williams	Hola les lolos
	Bob Williams	O Fracasso Não É o Fim
	Charlie Brown	I Know You (feat. Bastille) - Vigiland Remix
	Charlie Brown	End of The Day
	Charlie Brown	Pearl



Hướng phát triển trong tương lai

- Hệ thống gợi ý AI: Sử dụng dữ liệu lịch sử phát nhạc để xây dựng các thuật toán gợi ý thông minh, cá nhân hóa danh sách phát dựa trên sở thích và thói quen nghe nhạc.
- Phân tích dữ liệu người dùng: Đánh giá hành vi người dùng, như bài hát được nghe nhiều nhất, khung giờ nghe phổ biến, giúp tối ưu trải nghiệm cá nhân hóa.
- Quản lý xu hướng âm nhạc: Phân tích dữ liệu để phát hiện các xu hướng âm nhạc mới, hỗ trợ nghệ sĩ và nhà phát hành âm nhạc trong việc ra quyết định.
- Hỗ trợ marketing: Tạo các chiến dịch quảng cáo nhắm mục tiêu dựa trên sở thích âm nhạc hoặc mức độ tương tác của từng nhóm người dùng.



*

THANKS FOR LISTENING

