

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
VIỆN TRÍ TUỆ NHÂN TẠO

--- □ □ □ ---



BÁO CÁO MÔN CƠ SỞ DỮ LIỆU
ĐỀ TÀI

QUẢN LÝ THƯ VIỆN ÂM NHẠC

Giảng viên : TS Trần Hồng Việt
Sinh viên thực hiện :- Nguyễn Thị Ngọc Lan : 23020390
- Chu Thị Phương Anh : 23020324
- Phạm Hà Anh : 23020440
Lớp : INT2211 37

Hà Nội 12/2024

PHỤ LỤC

PHỤ LỤC.....	3
I. PHÁT BIỂU BÀI TOÁN.....	4
II. MÔ TẢ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU.....	4
III. MÔ HÌNH ER CỦA HỆ THỐNG.....	5
IV. MÔ HÌNH QUAN HỆ (CHUYỂN TỪ MÔ HÌNH ER).....	7
V. ĐẶC TẢ YÊU CẦU DỮ LIỆU (TỪ ĐIỂN DỮ LIỆU).....	8
VI. Danh sách các ràng buộc dữ liệu.....	14
VII. Cài đặt vật lý.....	14
A. INNER JOIN.....	14
B.OUTER JOIN.....	21
C.SUBQUERY IN WHERE.....	30
D.SUBQUERY IN FROM.....	39
F.TRANSACTIONS SỬ DỤNG ROLLBACK.....	50
G.TRIGGER.....	53
H.PROCEDURES.....	58

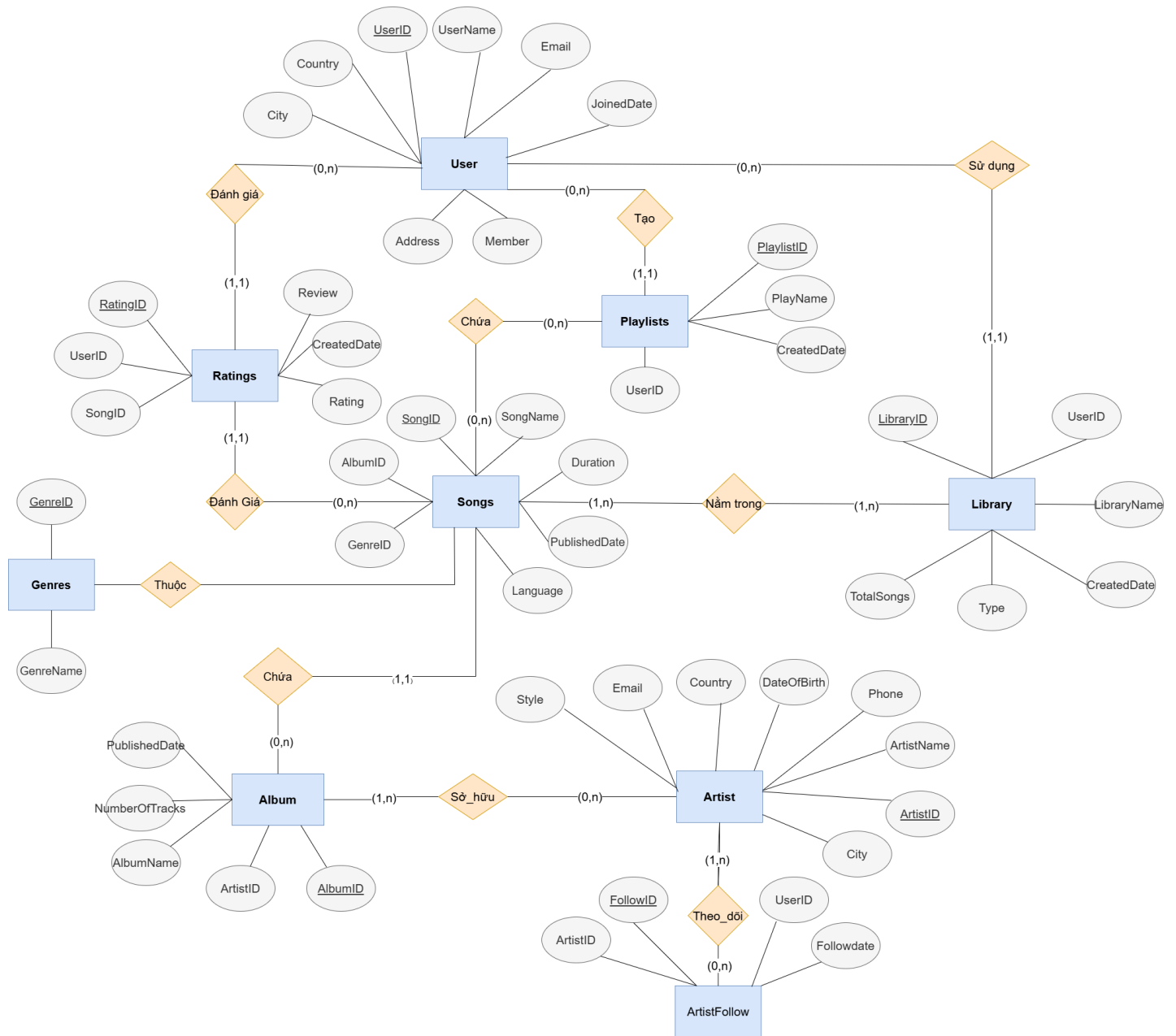
I. PHÁT BIỂU BÀI TOÁN

Cơ sở dữ liệu quản lý bài hát được tạo ra với mục đích quản lý và tổ chức thông tin liên quan đến các bài hát, nghệ sĩ, album và người dùng một cách hiệu quả, hỗ trợ quá trình tìm kiếm, lưu trữ và phân tích dữ liệu trong lĩnh vực âm nhạc.

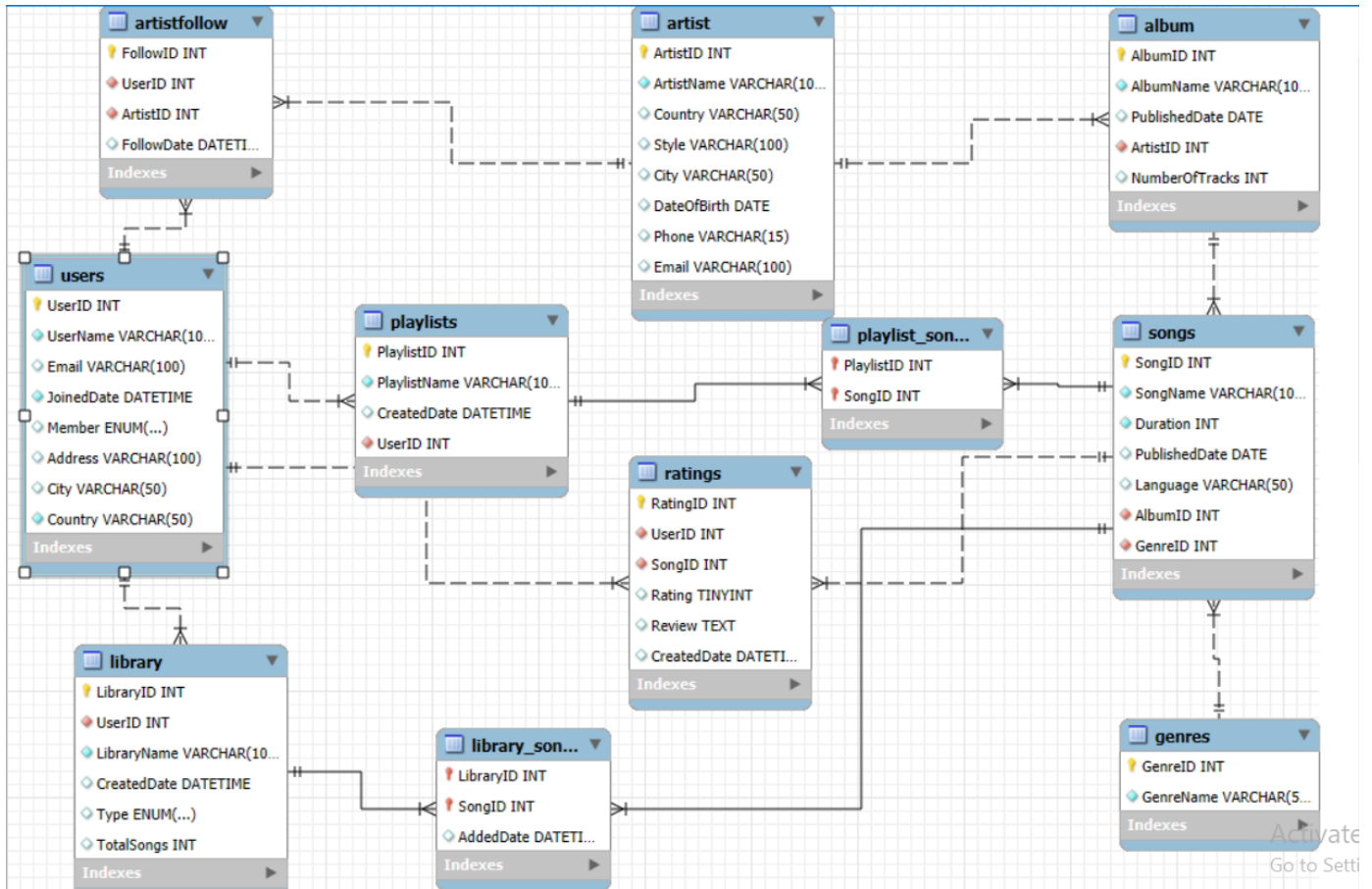
II. MÔ TẢ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU

- 1. Quản lý bài hát:** Hệ thống quản lý thông tin của từng bài hát bao gồm: tên bài hát, nghệ sĩ trình bày, album liên quan, thời lượng, lời bài hát, và thể loại. Bài hát có thể thuộc nhiều danh sách phát hoặc xuất hiện trong các album khác nhau. Bài hát có thể được tìm kiếm, thêm, chỉnh sửa hoặc xóa khi cần thiết
- 2. Quản lý nghệ sĩ:** Hệ thống lưu trữ thông tin chi tiết về các nghệ sĩ, bao gồm: tên, ngày sinh, quốc tịch, và các album, bài hát đã phát hành. Thông tin nghệ sĩ có thể được cập nhật, thêm mới hoặc xóa.
- 3. Quản lý album:** Mỗi album trong hệ thống được quản lý với các thông tin: tên album, nghệ sĩ, ngày phát hành, thể loại, và danh sách bài hát trong album. Hệ thống hỗ trợ phân loại album theo thể loại hoặc nghệ sĩ để người dùng dễ dàng tìm kiếm
- 4. Quản lý người dùng:** Hệ thống lưu trữ thông tin tài khoản người dùng, bao gồm: tên, email, mật khẩu, và các thông tin cá nhân khác. Người dùng có thể tạo Playlist, Library của riêng mình tùy theo sở thích cá nhân, chọn danh sách phát,...
- 5. Quản lý danh sách phát:** Người dùng có thể tạo các danh sách phát cá nhân với tên và mô tả riêng. Hệ thống cho phép thêm, xóa, hoặc sắp xếp các bài hát trong danh sách phát.
- 6. Quản lý thư viện cá nhân:** Mỗi người dùng có một thư viện cá nhân để lưu các bài hát, album, hoặc danh sách phát yêu thích. Thư viện cá nhân cho phép người dùng xem lại hoặc quản lý các nội dung đã lưu. Hỗ trợ đồng bộ thư viện trên nhiều thiết bị của cùng một người dùng

III. MÔ HÌNH ER CỦA HỆ THỐNG



IV. MÔ HÌNH QUAN HỆ (CHUYỂN TỪ MÔ HÌNH ER)



- **ARTISTFOLLOW** (FollowID, UserID, ArtistID, FollowDate)
- **USER** (UserID, UserName, Email, JoinedDate, Member, Address, City, Contry)
- **LIBRARY** (LibraryID, UserID, LibraryName, CreatedDate, Type, TotalSongs)
- **ARTIST** (ArtistID, ArtistName, Country, Style, City, DateOfBirth, Phone, Email)
- **PLAYLISTS** (PlaylistID, PlaylistName, CreatedDate, UserID)
- **LIBRARY_SONGS** (LibraryID, SongID, AddedDate)
- **RATINGS** (RatingID, UserID, SongID, Rating, Review, CreatedDate)
- **ALBUM** (AlbumID, AlbumName, PublishedDate, ArtistID, NumberOfTracks)
- **SONGS** (SongID, SongName, Duration, PublishedDate, Language, AlbumID, GenreID)
- **GENRES** (GenreID, GenreName)

V. ĐẶC TẢ YÊU CẦU DỮ LIỆU (TỪ ĐIỂN DỮ LIỆU)

Data Element	Description	Composition of Data Type	Length	Values
UserID (Users)	Unique identifier for a user	Integer	Variable	e.g., 1,2,3
UserName (Users)	Name of the user	String (Text)	100 characters	e.g., 'John', 'Anna',...
Email (Users)	Email of the user	String (Text)	100 characters	e.g., johndoe1@example.com, janesmith2@example.com, alicejohnson3@example.com,...
JoinedDate (Users)	Date of joining	Date	Standard Date Format (e.g., YYYY-MM-DD)	e.g., '2021-09-25', '2022-11-23', '2024-03-04',...
Member (Users)	Type of member (Vip, casual, standard,...)	String (Text)	100 characters	e.g., 'NORMAL', 'PREMIUM', 'VIP',...
Address (Users)	The address of the user	String (Text)	100 characters	e.g., '10 Downing St', '1 Harbour Rd', '456 Opera House Rd',...
City (Users)	City of the user	String (Text)	50 characters	e.g., 'London', 'Sydney', 'Tokyo',...
Country (Users)	Country of the user	String (Text)	50 characters	e.g., 'UK', 'Australia', 'Germany',...

FollowID (ArtistFollow)	Unique identifier for following	Integer	Variable	e.g., 1,2,3,4,..
UserID (ArtistFollow)	Unique identifier for a user, serving as a foreign key referencing the Users table.	Integer	Variable	e.g., 11,12,23,43,..
ArtistID (ArtistFollow)	Unique identifier for an artist, serving as a foreign key referencing the Artists table.	Integer	Variable	e.g., 1,2,3,4,...
FollowDate (ArtistFollow)	Date of following	Date	Standard Date Format (e.g., YYYY-MM-DD)	e.g., '2021-09-25', '2022-11-23', '2024-03-04',...
LibraryID (Library)	Unique identifier for library	Integer	Variable	e.g., 1,2,3,4,...
UserID (Library)	Unique identifier for a user	Integer	Variable	e.g., 1,2,3,4,..
LibraryName (Library)	Name of Library	String (Text)	100 characters	e.g., 'Chill & Relax ', 'Top 100 Charts ', 'Hip Hop & R&B ',...
CreatedDate (Library)	Date of creating	Date	Standard Date Format (e.g.,	e.g., '2021-09-25', '2022-11-23',

			YYYY-MM-DD)	'2024-03-04',...
Type (Library)	Type of library	String (Text)	100 characters	e.g.,
TotalSongs (Library)	Number of total songs	Integer	Variable	e.g., 100,24, 34,...
ArtistID (Artist)	Unique identifier for an Artist	Integer	Variable	e.g., 1,2,3,4,...
ArtistName (Artist)	Name of the Artist	String (Text)	100 characters	e.g., 'Taylor Swift', 'Ed Sheeran',....
Country (Artist)	Country of the Artist	String (Text)	50 characters	e.g., 'USA', 'UK',...
Style (Artist)	Styles of the Artist	String (Text)	100 characters	e.g., 'Pop', 'Pop, Folk', 'K-Pop',....
City (Artist)	City of the Artist	String (Text)	50 characters	e.g., 'Nashville', 'Seoul', 'London',...
DateOfBirth (Artist)	When an artist was born	Date	Standard Date Format (e.g., YYYY-MM-DD)	e.g., '1989-12-13', '1991-02-17',....
Phone (Artist)	Phone Number of the Artist	String (Text)	15 characters	e.g., '1234567890', '0987654321',...
Email (Artist)	Email of the Artist	String (Text)	100 characters	e.g., 'taylor@swift.com', 'ed@sheeran.com', ..
PlaylistID (Playlists)	Unique identifer for Playlists	Integer	Variable	e.g., 1,2,3,4,5,...
PlaylistName	Name of the	String	100 characters	e.g., 'The Ultimate

(Playlists)	Playlist	(Text)		Rock Classics Playlist for Every Mood 836’, ‘Chill Vibes for a Perfect Evening with Friends 713’,...
CreatedDate (Playlists)	The created date of a Playlist	Date	Standard Date Format (e.g., YYYY-MM-DD)	e.g., ‘2021-02-06 00:00:00’, ‘2024-02-01 00:00:00’,....
UserID (Playlists)	Unique identifier for an User, serving as a foreign key referencing the User table.	Integer	Variable	e.g, 33, 51,...
LibraryID (Library_songs)	Unique identifier for a pair of the (Library and Song)	Integer	Variable	e.g, 1,2,3,4,....
SongID (Library_songs)	Unique identifier for a song, erving as a foreign key referencing the Songs table.	Integer	Variable	e.g., 23, 389,...
AddedDate (Library_songs)	The date that song was added	Date	Standard Date Format (e.g., YYYY-MM-DD)	e.g., ‘2024-10-03 00:00:00’, ‘2024-11-21 00:00:00’,...
RatingID (Ratings)	Unique identifier for a rating instance	Integer	Variable	e.g, 1,2,3,....
UserID (Ratings)	Unique identifier for a rating instance,	Integer	Variable	e.g., 68,146,...

	serving as a foreign key referencing the Users table.			
SongID (Ratings)	Unique identifier for a Songs, serving as a foreign key referencing the Songs table.	Integer	Variable	e.g., 356, 255,...
Review (Ratings)	Review of a song was given by an User	String (Text)	65,535 characters	e.g., 'Great Song!', 'Amazing Song!',....
CreatedDate (Ratings)	Date of creating	Date	Standard Date Format (e.g., YYYY-MM-DD)	e.g., '2024-12-14 16:58:05', '2010-10-16 00:00:00',...
AlbumID (Album)	Unique identifier for an Album	Integer	Variable	e.g., 1,2,3,4,..
AlbumName (Album)	Name of the Album	String (Text)	100 characters	e.g., 'Map of the Soul: 7', 'BE',...
PublishedDate (Album)	Date of publishing	Date	Standard Date Format (e.g., YYYY-MM-DD)	e.g., '2020-02-21', '2018-08-24',...
ArtistID (Album)	Unique identifier for an Artist, serving as a foreign key referencing the Artists table.	Integer	Variable	e.g., 1, 2, 3,....
NumberOfTracks (Album)	Number of Tracks of an Album	Integer	Variable	e.g, 20, 8, 26,...

SongID (Songs)	Unique identifier for a song	Integer	Variable	e.g, 1, 2, 3,...
SongName (Songs)	Name of the song	String (Text)	100 characters	e.g., 'Intro: Boy Meets Evil', 'Blood Sweat & Tears',....
Duration (Songs)	Duration of the song	Integer	Variable	e.g, 229, 325,...
PublishedDate (Songs)	Date of publishing a song	Date	Standard Date Format (e.g., YYYY-MM-DD)	e.g., '2018-08-24', '2020-02-21',....
Language (Songs)	The primary language used in a song	String (Text)	50 characters	e.g., 'Korean', 'English',...
AlbumID (Songs)	Unique identifier for an Album, serving as a foreign key referencing the Album table	Integer	Variable	e.g., 1,2 ,3,4,5,...
GenreID (Songs)	Unique identifier for a Genre, serving as a foreign key referencing the Genres table	Integer	Variable	e.g, 1,2 ,3,....
GenreID (Genres)	Unique identifier for a Genre	Integer	Variable	e.g, 1,2 ,3,...
GenreName (Genres)	Name of a Genre	String (Text)	50 characters	e.g., 'Acid Jazz', 'Acoustic',....

VI. Danh sách các ràng buộc dữ liệu

- **User và Ratings:** Mỗi quan hệ qua trường **UserID**. Mỗi người dùng (User) có thể đánh giá nhiều bài hát (Ratings), nhưng mỗi đánh giá thuộc về một người dùng duy nhất.
- **Songs và Ratings:** Mỗi quan hệ qua trường **SongID**. Mỗi bài hát (Song) có thể nhận được nhiều đánh giá (Ratings), nhưng mỗi đánh giá thuộc về một bài hát duy nhất.
- **Genres và Songs:** Mỗi quan hệ qua trường **GenreID**. Mỗi bài hát thuộc về một thể loại (Genres), và một thể loại có thể chứa nhiều bài hát.
- **Album và Songs:** Mỗi quan hệ qua trường **AlbumID**. Một album có thể chứa nhiều bài hát (Songs), nhưng mỗi bài hát chỉ thuộc về một album duy nhất.
- **Artist và Album:** Mỗi quan hệ qua trường **ArtistID**. Một nghệ sĩ (Artist) có thể sở hữu nhiều album, nhưng mỗi album chỉ thuộc về một nghệ sĩ.
- **Artist và ArtistFollow:** Mỗi quan hệ qua trường **ArtistID**. Một nghệ sĩ (Artist) có thể được nhiều người dùng theo dõi, nhưng mỗi lượt theo dõi chỉ gắn với một nghệ sĩ.
- **User và ArtistFollow:** Mỗi quan hệ qua trường **UserID**. Một người dùng có thể theo dõi nhiều nghệ sĩ (Artist), nhưng mỗi lượt theo dõi chỉ thuộc về một người dùng.
- **Library và User:** Mỗi quan hệ qua trường **UserID**. Một người dùng (User) có thể sở hữu nhiều thư viện (Library), nhưng mỗi thư viện thuộc về một người dùng.
- **Library và Songs:** Mỗi quan hệ qua trường **SongID**. Một thư viện có thể chứa nhiều bài hát (Songs), và một bài hát có thể nằm trong nhiều thư viện.
- **User và Playlists:** Mỗi quan hệ qua trường **UserID**. Một người dùng có thể tạo nhiều danh sách phát (Playlists), nhưng mỗi danh sách phát chỉ thuộc về một người dùng.
- **Playlists và Songs:** Mỗi quan hệ qua trường **SongID**. Một danh sách phát (Playlists) có thể chứa nhiều bài hát (Songs), và một bài hát có thể nằm trong nhiều danh sách phát.

VII. Cài đặt vật lý

A. INNER JOIN

1. Lấy danh sách các bài hát cùng với tên nghệ sĩ của chúng

```
SELECT s.SongName, a.ArtistName
FROM Songs s
INNER JOIN Album al ON s.AlbumID = al.AlbumID
INNER JOIN Artist a ON al.ArtistID = a.ArtistID;
```

	SongName	ArtistName
►	Just Dance	Taylor Swift
	Lovegame	Taylor Swift
	Paparazzi	Taylor Swift
	Poker Face	Taylor Swift
	Eh, Eh (Nothing Else I Can Say)	Taylor Swift
	Beautiful, Dirty, Rich	Taylor Swift
	Bad Romance	Taylor Swift
	Alejandro	Taylor Swift
	Monster	Taylor Swift
	Speechless	Taylor Swift
	Dance In The Dark	Taylor Swift
	Supermodel	Taylor Swift
	Love Galore	Taylor Swift
	Drew Barrymore	Taylor Swift

2. Thông tin bài hát cùng ngôn ngữ và thể loại

```
SELECT Songs.SongName, Songs.Language, Genres.GenreName
FROM Songs
INNER JOIN Genres ON Songs.GenreID = Genres.GenreID;
```

	SongName	Language	GenreName
►	Intro: Boy Meets Evil	Korean	K-pop
	Blood Sweat & Tears	Korean	K-pop
	Begin	Korean	K-pop
	Lie	Korean	K-pop
	Stigma	Korean	K-pop
	First Love	Korean	K-pop
	Reflection	Korean	K-pop
	Mama	Korean	K-pop
	Blue & Grey	Korean	K-pop
	Telepathy	Korean	K-pop
	Stay	Korean	K-pop
	Dynamite	English	K-pop

3. Truy vấn danh sách album kèm tên bài hát sao cho Album đó có ít nhất 10 bài hát có Genre là 'Country'

```
SELECT Album.AlbumName, Songs.SongName, Genres.GenreName
FROM Album
JOIN Songs on Album.AlbumID= Songs.AlbumID
JOIN Genres on Songs.GenreID = Genres.GenreID
where Genres.GenreName = 'Country';
```

	AlbumName	SongName	GenreName
▶	Everyday Life	Slow Burn	Country
	Everyday Life	I Miss You	Country
	Everyday Life	Butterflies	Country
	Everyday Life	Love Is a Wild Thing	Country
	Everyday Life	Space Cowboy	Country
	Everyday Life	Happy & Sad	Country
	Everyday Life	Mother	Country
	Everyday Life	Oh, What a World	Country
	Everyday Life	Golden Hour	Country
	Everyday Life	Teenage Wildflower	Country
	Everyday Life	Wonder Woman	Country

4. Truy vấn tìm ra danh sách các Artist có người theo dõi là ở 'Chicago'

```
select a.ArtistName, u.UserName, u.City
from Artist a
join ArtistFollow af on af.ArtistID = a.ArtistID
join Users u on u.UserID = af.UserID
where u.City = 'Chicago';
```

	ArtistName	UserName	City
▶	Halsey	Tina White	Chicago
	Faye Wong	Gabriella Silver	Chicago
	Kygo	Olivia Rose	Chicago
	Alan Walker	Yara Grey	Chicago
	Beyoncé	Jack Yellow	Chicago
	Jay Chou	Sophia Yellow	Chicago

5. Truy vấn danh sách thư viện của người dùng tên 'Rose' và tổng số bài hát trong từng thư viện

```
SELECT u.UserName,
       lb.LibraryName,
       COUNT(ls.SongID) AS TotalSongs
FROM Users u
JOIN Library lb ON u.UserID = lb.UserID
JOIN Library_Songs ls ON lb.LibraryID = ls.LibraryID
WHERE u.UserName like '%Rose%'
GROUP BY u.UserName, lb.LibraryName;
```

	UserName	LibraryName	TotalSongs
▶	Tom Rose	Hip Hop & R&B ...	1
	Ethan Rose	90s Classics 158	4

6. Truy vấn danh sách các nghệ sĩ có album phát hành trong năm 2020 và thuộc thể loại 'Pop'

```
SELECT a.ArtistName,
       al.AlbumName,
```

```

g.GenreName
FROM Artist a
JOIN Album al ON a.ArtistID = al.ArtistID
JOIN Songs s ON al.AlbumID = s.AlbumID
JOIN Genres g ON s.GenreID = g.GenreID
WHERE g.GenreName = 'Pop' AND YEAR(al.PublishedDate) = 2020;

```

	ArtistName	AlbumName	GenreName
▶	The Weeknd	After Hours	Pop
	The Weeknd	After Hours	Pop

7. Thể loại ở USA có rating cao nhất

```

SELECT g.GenreName, AVG(r.Rating) AS AvgRating
FROM genres g
INNER JOIN songs s ON g.GenreID = s.GenreID
INNER JOIN ratings r ON s.SongID = r.SongID
INNER JOIN users u ON r.UserID = u.UserID
WHERE u.Country = 'USA'
GROUP BY g.GenreName
ORDER BY AvgRating DESC
LIMIT 1;

```

	GenreName	AvgRating
▶	House	4.0000
	Dancehall	3.5000
	Lo-fi	3.5000
	Jazz	3.4000
	Country	3.3333
	K-pop	2.8571
	Pop	2.7222
	R&B	2.2941
	Rock	2.0000

8. Thống kê số lượng bài hát của mỗi nghệ sĩ được đánh giá trung bình trên 4 theo từng năm phát hành

```

SELECT a.ArtistName, YEAR(s.PublishedDate) AS ReleaseYear,
COUNT(s.SongID) AS TotalHighRatedSongs
FROM artist a
INNER JOIN album al ON a.ArtistID = al.ArtistID
INNER JOIN songs s ON al.AlbumID = s.AlbumID
INNER JOIN ratings r ON s.SongID = r.SongID
GROUP BY a.ArtistID, YEAR(s.PublishedDate)
HAVING AVG(r.Rating) > 4
ORDER BY ReleaseYear ASC, TotalHighRatedSongs DESC;



```

	ArtistName	ReleaseYear	TotalHighRatedSongs
▶	Ozuna	2012	1
	G.E.M.	2015	2
	BTS	2015	1
	Jay Chou	2019	2

B.OUTER JOIN

1. Liệt kê tên nghệ sĩ và số lượng Album bao gồm cả những người không có Album nào

`select a.ArtistName, a.ArtistID, count(al.AlbumID) as NumberOfAlbums from Artist a
left join Album al on a.ArtistID = al.ArtistID
group by a.ArtistName, a.ArtistID;`

Result Grid			 Filter Rows:	
	ArtistName	ArtistID	NumberOfAlbums	
▶	Taylor Swift	1	3	
	Ed Sheeran	2	2	
	BTS	3	5	
	Adele	4	1	
	Coldplay	5	2	
	Beyoncé	6	2	
	Drake	7	2	
	Lady Gaga	8	3	
	The Weeknd	9	2	
	Imagine Dragons	10	2	
	Bruno Mars	11	2	
	Post Malone	12	1	
	Ariana Grande	13	2	
	Billie Eilish	14	1	
	Shawn Mendes	15	1	
	Rihanna	16	5	

2. Liệt kê tất cả Users và số lượng Library họ đã tạo bao gồm cả những người không có Library nào

`select u.UserName, u.UserID, count(l.UserID) as NumberOfLibrarys from Users u
left join Library l on u.UserID = l.UserID group by u.UserName, u.UserID;`

	UserName	UserID	NumberOfLibrarys
▶	Michael Green	1	1
	Emily Davis	2	0
	Sofia Nguyen	3	0
	David Lee	4	1
	Sophia Martinez	5	2
	David Green	6	2
	Eva Black	7	1
	Frank Blue	8	1
	Grace Red	9	0
	Henry Gold	10	0
	Ivy Silver	11	2
	Jack Rose	12	1
	Kathy Yellow	13	2
	Leo Grey	14	0
	Mona Purple	15	2
	Nathan Pink	16	1

3. Danh sách tất cả các playlist và điểm đánh giá trung bình của các bài hát trong playlist đó

```
SELECT p.PlaylistName, COALESCE(AVG(r.Rating), 0) AS AverageRating
FROM playlists p
LEFT OUTER JOIN playlist_songs ps ON p.PlaylistID = ps.PlaylistID
LEFT OUTER JOIN ratings r ON ps.SongID = r.SongID
GROUP BY p.PlaylistName;
```

PlaylistName	AverageRating
The Ultimate Rock Classics Playlist for Every Mo...	2.6667
Chill Vibes for a Perfect Evening with Friends 713	2.0000
Essential Jazz Tracks to Relax and Unwind 894	3.5000
Summer Hits to Keep You Dancing All Day 142	4.0000
Indie Beats to Enhance Your Creative Mindset 451	3.3333
Party Playlist for an Unforgettable Night Out 274	0.0000
Relaxing Tunes to Calm Your Soul After a Busy ...	3.5000
Workout Motivation with High-Energy Songs 852	2.2500
Classical Masterpieces to Set the Perfect Mood ...	3.5000
Throwback Hits from the 90s to Relive Your You...	3.0000
Mood Boosters to Lift Your Spirits in Any Situati...	5.0000
Romantic Playlist for a Cozy Night with Someon...	3.0000
Morning Coffee and Chill Tunes to Start Your Da...	0.0000
Dance Party Playlist for the Best Moves All Nigh...	2.5000
Rock and Roll Anthems to Pump Up Your Energy...	3.0000
Acoustic Favorites for a Cozy Afternoon at Hom...	3.5000
Hip Hop Beats to Feel the Rhythm of the Street...	2.5000
Relaxing Piano Melodies for a Peaceful Evening ...	0.0000
Pop Hits for Every Celebration and Good Time 930	0.0000
Road Trip Tunes for Your Next Adventure 168	0.0000

4. Tìm nghệ sĩ không có bài hát nào được người dùng Rachel Cyan đánh giá

```
select a.ArtistName, count(r.SongID), count(u.Member) from Artist a
left join Album al on a.ArtistID = al.ArtistID
left join songs s on al.AlbumID = s.AlbumID
left join Ratings r on s.SongID = r.SongID
left join Users u on r.UserID = u.UserID and u.Member = 'Rachel Cyan'
group by a.ArtistName
having count(r.SongID) = 0 or count(u.Member) = 0;
```

	ArtistName	count(r.SongID)	count(u.Member)
▶	Taylor Swift	10	0
	Ed Sheeran	8	0
	BTS	14	0
	Adele	4	0
	Coldplay	9	0
	Beyoncé	3	0
	Drake	6	0
	Lady Gaga	5	0
	The Weeknd	3	0
	Imagine Dragons	5	0
	Bruno Mars	5	0
	Post Malone	1	0
	Ariana Grande	3	0
	Billie Eilish	2	0
	Shawn Mendes	4	0
	Rihanna	12	0
	Maroon 5	2	0
	Katy Perry	2	0
	Justin Bieber	5	0

5. Truy vấn danh sách người dùng và các playlist họ tạo, bao gồm cả người dùng không tạo playlist nào

```
SELECT u.UserName,
       p.PlaylistName
FROM Users u
LEFT JOIN Playlists p ON u.UserID = p.UserID;
```

	UserName	PlaylistName
▶	Michael Green	Synthwave Hits to Get Lost in Time 501
	Michael Green	Dream Pop Playlist for Peaceful Nights 203
	Michael Green	Electronic Chillout for a Modern Relaxation 248
	Michael Green	Psybient Beats for Relaxed Energy
	Emily Davis	NULL
	Sofia Nguyen	Summer Chill: The Ultimate Beach Playlist
	David Lee	NULL
	Sophia Martinez	Ambient Sounds for a Perfect Meditation Experi...
	David Green	NULL
	Eva Black	NULL
	Frank Blue	Alternative Rock Gems for a Fresh Sound 302
	Frank Blue	R&B Throwbacks for the Soul
	Grace Red	NULL
	Henry Gold	NULL
	Ivy Silver	NULL
	Jack Rose	Vocal Jazz for Relaxing Moments 586
	Kathy Yellow	NULL
	Leo Grey	NULL
	Mona Purple	Neo-Soul for a Modern Groove

6. Danh sách tất cả các nghệ sĩ cùng số lượng bài hát đã phát hành (kể cả nghệ sĩ chưa có bài hát).

```
SELECT a.ArtistName, COUNT(s.SongID) AS SongCount
FROM artist a
LEFT OUTER JOIN album ab ON ab.ArtistID = a.ArtistID
left outer join songs s on s.albumID = ab.albumID
GROUP BY a.ArtistName;
```

	ArtistName	SongCount
▶	Taylor Swift	33
	Ed Sheeran	24
	BTS	88
	Adele	6
	Coldplay	21
	Beyoncé	13
	Drake	21
	Lady Gaga	28
	The Weeknd	16
	Imagine Dragons	25
	Bruno Mars	13
	Post Malone	11
	Ariana Grande	13
	Billie Eilish	11
	Shawn Mendes	11
	Rihanna	39
	Maroon 5	11
	Katy Perry	11
	Justin Bieber	15

7. Tất cả bài hát và tổng số người dùng đã thêm bài hát đó vào thư viện

```

SELECT s.SongName, COUNT(l.UserID) AS TotalUsers
FROM songs s
LEFT OUTER JOIN library_songs ls ON s.SongID = ls.SongID
LEFT OUTER JOIN Library l on l.libraryID = ls.libraryID
GROUP BY s.SongID, s.SongName;

```

	SongName	TotalUsers
	X Marks the Spot	0
	Pray You Catch Me	0
	Hold Up	0
	Don't Hurt Yourself	0
	Sorry	0
	6 Inch	0
	Daddy Lessons	0
	Love Drought	1
	Sandcastles	0
	Forward	1
	Freedom	0
	All Night	0
	Formation	1
	Formula of Love	0
	The Feels	0
	Perfect Love	0
	Chemistry	0
	Butterfly	1
	Kiss Me	0

C.SUBQUERY IN WHERE

1. Tìm các nghệ sĩ có ít nhất một bài hát được đánh giá

```
select * from Artist
where ArtistID in
( select al.ArtistID from Album al
left join songs s on al.AlbumID = s.AlbumID
left join Ratings r on s.SongID = r.SongID );
```

	ArtistID	ArtistName	Country	Style	City	DateOfBirth	Phone	Email
▶	1	Taylor Swift	USA	Pop	Nashville	1989-12-13	1234567890	taylor@swift.com
	2	Ed Sheeran	UK	Pop, Folk	Halifax	1991-02-17	0987654321	ed@sheeran.com
	3	BTS	South Korea	K-pop	Seoul	2013-06-13	5678901234	bts@bigent.com
	4	Adele	UK	Pop, Soul	London	1988-05-05	8765432109	adele@music.com
	5	Coldplay	UK	Alternative Rock	London	1996-01-01	1230984567	coldplay@band.com
	6	Beyoncé	USA	R&B, Pop	Houston	1981-09-04	3216549870	beyonce@queenb.com
	7	Drake	Canada	Hip-Hop, R&B	Toronto	1986-10-24	4567890123	drake@ovo.com
	8	Lady Gaga	USA	Pop	New York	1986-03-28	9876543210	gaga@monster.com
	9	The Weeknd	Canada	R&B, Pop	Toronto	1990-02-16	6543217890	weeknd@starboy.com
	10	Imagine Dragons	USA	Alternative Rock	Las Vegas	2008-01-01	2345678901	idragons@music.com
	11	Bruno Mars	USA	Pop, Funk	Honolulu	1985-10-08	8901234567	bruno@mars.com
	12	Post Malone	USA	Hip-Hop, Pop	Syracuse	1995-07-04	1234567809	post@malone.com
	13	Ariana Grande	USA	Pop, R&B	Boca Raton	1993-06-26	0987612345	ariana@grande.com
	14	Billie Eilish	USA	Pop, Alternative	Los Angeles	2001-12-18	5678901235	billie@eilish.com
	15	Shawn Mendes	Canada	Pop, Folk	Pickering	1998-08-08	7654321098	shawn@mendes.com
	16	Rihanna	Barbados	Pop, R&B	Bridgetown	1988-02-20	2345678091	rihanna@fenty.com
	17	Maroon 5	USA	Pop Rock	Los Angeles	2001-06-01	3219876540	maroon5@band.com
	18	Katy Perry	USA	Pop	Santa Ba...	1984-10-25	6549873210	katy@perry.com
	19	Justin Bieber	Canada	Pop, R&B	Stratford	1994-03-01	7890123456	justin@bieber.com

2. Tìm các bài hát không thuộc thể loại nào đang phổ biến (có trên 50 bài hát)

```
SELECT * FROM Songs
WHERE GenreID NOT IN (
    SELECT GenreID
    FROM Songs
    GROUP BY GenreID
    HAVING COUNT(SongID) > 50
);
```

	SongID	SongName	Duration	PublishedDate	Language	AlbumID	GenreID
▶	96	Secret	242	2003-07-31	English	27	13
	97	Forever	251	2003-07-31	English	27	13
	98	Love Me	263	2003-07-31	English	27	13
	99	Night	215	2003-07-31	English	27	13
	100	Believe	238	2003-07-31	English	27	13
	101	I Want You	229	2003-07-31	English	27	13
	102	Unspoken	257	2003-07-31	English	27	13
	103	Angel	231	2003-07-31	English	27	13
	104	Faith	245	2003-07-31	English	27	13
	105	Perfect	240	2003-07-31	English	27	13
	106	True Love	252	2003-07-31	English	27	13
	107	Shining Star	249	2003-07-31	English	27	13
	204	Slow Burn	245	2018-03-30	English	42	8
	205	I Miss You	222	2018-03-30	English	42	8
	206	Butterflies	220	2018-03-30	English	42	8
	207	Love Is a ...	233	2018-03-30	English	42	8
	208	Space Co...	276	2018-03-30	English	42	8
	209	Happy & Sad	229	2018-03-30	English	42	8
	210	Mother	258	2018-03-30	English	42	8

3. Tìm người dùng chỉ theo dõi nghệ sĩ từ một quốc gia duy nhất

```
SELECT UserName FROM Users
WHERE UserID IN (
    SELECT af.UserID FROM ArtistFollow af
    JOIN Artist a ON af.ArtistID = a.ArtistID
    GROUP BY af.UserID
    HAVING COUNT(DISTINCT a.Country) = 1
);
```

	UserName
▶	Michael Green
	Sofia Nguyen
	David Lee
	Sophia Martinez
	David Green
	Eva Black
	Frank Blue
	Grace Red
	Henry Gold
	Ivy Silver
	Jack Rose
	Kathy Yellow
	Leo Grey
	Mona Purple
	Nathan Pink
	Olivia Cyan
	Paul Orange
	Quincy Brown
	Rachel Indigo

4. Tìm tất cả bài hát được xuất bản sau album có ngày phát hành sớm nhất

```
select * from Songs
```

where PublishedDate > (select min(PublishedDate) from Album);

	SongID	SongName	Duration	PublishedDate	Language	AlbumID	GenreID
▶	1	Intro: Boy Meets Evil	120	2016-10-10	Korean	4	4
	2	Blood Sweat & Tears	217	2016-10-10	Korean	4	4
	3	Begin	231	2016-10-10	Korean	4	4
	4	Lie	217	2016-10-10	Korean	4	4
	5	Stigma	218	2016-10-10	Korean	4	4
	6	First Love	230	2016-10-10	Korean	4	4
	7	Reflection	233	2016-10-10	Korean	4	4
	8	Mama	218	2016-10-10	Korean	4	4
	9	Blue & Grey	255	2020-11-20	Korean	2	4
	10	Telepathy	202	2020-11-20	Korean	2	4
	11	Stay	205	2020-11-20	Korean	2	4
	12	Dynamite	199	2020-11-20	English	2	4
	13	Mine	230	2010-08-04	English	38	1
	14	Back to December	290	2010-11-15	English	38	1
	15	Speak Now	244	2010-10-25	English	38	1
	16	Euphoria	229	2018-08-24	Korean	3	4
	17	Trivia: Just Dance	225	2018-08-24	Korean	3	4
	18	Serendipity	276	2018-08-24	Korean	3	4
	19	DNA	223	2018-08-24	Korean	3	4

5. Tìm các bài hát có tổng điểm đánh giá thấp hơn điểm trung bình của tất cả các bài hát

```
select SongName from Songs
where SongID in (select SongID from Ratings
group by SongID
having sum(Rating) < (select avg(sum_Rating) from (
select sum(Rating) as sum_Rating from Ratings
group by SongID) as Sub )
);
```

	SongName
▶	First Love
	DNA
	Pray You Catch Me
	Sandcastles
	Kiss Me
	The Last
	Tempo
	No Way
	Palette
	I Like You
	I Miss You
	Love Me
	Believe
	Angel
	Miracle
	Home
	Dreamer
	The Story of Us
	Dangerous Woma...

6. Truy vấn tên nghệ sĩ có bài hát được đánh giá thấp nhất trong hệ thống (dựa trên cột Rating).

```
SELECT a.ArtistName
FROM Artist a
```

```

WHERE a.ArtistID IN (
  SELECT al.ArtistID
  FROM Album al join Songs s on s.AlbumID = al.AlbumID
  JOIN Ratings r ON s.SongID = r.SongID
  WHERE r.Rating = (
    SELECT MIN(r_sub.Rating)
    FROM Ratings r_sub
  ));

```

ArtistName
Taylor Swift
BTS
Adele
Beyoncé
Drake
The Weeknd
Imagine Dragons
Bruno Mars
Ariana Grande
Billie Eilish
Shawn Mendes
Rihanna
Maroon 5
Selena Gomez
Dua Lipa
BLACKPINK
TWICE
EXO
Jay Chou

7. Truy vấn danh sách thư viện có chứa bài hát của nghệ sĩ đến từ quốc gia 'Canada'.

```

select lb.LibraryName,
       s.SongName
  from Library lb join Library_songs ls on ls.LibraryID = lb.LibraryID
 join songs s on s.SongID = ls.SongID
 where s.SongID in
 (select s.SongID from songs
  join album a on a.AlbumID = s.AlbumID
  join artist art on art.ArtistID = a.ArtistID
  where art.Country = 'Canada');

```

LibraryName	SongName
Top 100 Charts 677	Staring at the Sun
Hip Hop & R&B 27	Enemies
Electronic Beats 453	People You Know
Electronic Beats 102	Falling
Top 100 Charts 438	Dance Again
Summer Vibes 382	Die for Me
Electronic Beats 643	Your Power
Workout Power 343	Hollywoods Bleeding
Hip Hop & R&B 470	Getting Older
Chill & Relax 588	Let Me Get Me
Personal Playlist 88	Cut You Off
Top 100 Charts 515	People You Know

8. Các bài hát trong album có tên bắt đầu bằng 'A'

```
SELECT *
FROM songs
WHERE AlbumID IN (
    SELECT AlbumID
    FROM album
    WHERE AlbumName LIKE 'A%'
);
```

SongID	SongName	Duration	PublishedDate	Language	AlbumID	GenreID
183	Believer	204	2017-06-23	English	7	2
184	Thunder	210	2017-06-23	English	7	2
185	Whatever It Takes	215	2017-06-23	English	7	2
186	I Don't Know Why	220	2017-06-23	English	7	2
187	Mouth of the River	225	2017-06-23	English	7	2
188	Start Over	230	2017-06-23	English	7	2
189	Dancing in the Dark	210	2017-06-23	English	7	2
190	Rise Up	218	2017-06-23	English	7	2
191	Walking the Wire	205	2017-06-23	English	7	2
192	The Fall	240	2017-06-23	English	7	2
152	Getting Older	220	2021-07-30	English	11	1
153	I Didn't Change M...	230	2021-07-30	English	11	1
154	Billie Bossa Nova	204	2021-07-30	English	11	1
155	My Future	210	2021-07-30	English	11	1
156	Oxytocin	222	2021-07-30	English	11	1
157	Goldwing	237	2021-07-30	English	11	1
163	Lost Cause	213	2021-07-30	English	11	1
164	Halo	219	2021-07-30	English	11	1
165	Everybody Dies	225	2021-07-30	English	11	1

D.SUBQUERY IN FROM

1. Tính số người dùng trung bình trên mỗi quốc gia (Subquery trong FROM)

```
SELECT Country, AVG(UserCount) AS AvgUsers
FROM (
    SELECT Users.Country, COUNT(Users.UserID) AS UserCount
    FROM Users
    GROUP BY Users.Country
```


) AS CountryUsers
GROUP BY Country;

Country	AvgUsers
UK	4.0000
Australia	4.0000
Japan	10.0000
Canada	4.0000
France	4.0000
USA	70.0000
Germany	3.0000
India	8.0000
South Africa	1.0000
Singapore	15.0000
China	21.0000
Thailand	11.0000
South Korea	10.0000
Indonesia	5.0000

2. Playlist chứa nhiều bài hát nhất

```
SELECT *
FROM (
    SELECT p.PlaylistName, COUNT(ps.SongID) AS TotalSongs
    FROM playlists p
    LEFT JOIN playlist_songs ps ON p.PlaylistID = ps.PlaylistID
    GROUP BY p.PlaylistName
) AS PlaylistCounts
ORDER BY TotalSongs DESC
LIMIT 1;
```

PlaylistName	TotalSongs
The Ultimate Rock Classics Playlist for Every Mo...	12

3. Truy vấn danh sách tên bài hát và tên nghệ sĩ cho các bài hát có ít nhất một đánh giá lớn hơn 3 sao từ người dùng ở 'Milan', 'Italy'

```
SELECT song_with_ratings.SongName, song_with_ratings.ArtistName
FROM (
    SELECT s.SongName, a.ArtistName, r.Rating
    FROM Songs s
    JOIN Album al ON al.AlbumID = s.AlbumID
    JOIN Artist a ON al.ArtistID = a.ArtistID
    JOIN Ratings r ON s.SongID = r.SongID
    JOIN Users u ON r.UserID = u.UserID
    WHERE u.City = 'Milan' AND r.Rating >= 3 AND u.Country = 'Italy'
) AS song_with_ratings;
```

SongName	ArtistName
A Thousand Bad Times	Drake

4. Truy vấn danh sách tên bài hát và album của bài hát có đánh giá cao nhất (5 sao)

```
SELECT song_with_ratings.SongName, song_with_ratings.AlbumName
```

```

FROM (
    SELECT s.SongName, a.AlbumName, r.Rating
    FROM Songs s
    JOIN Album a ON s.AlbumID = a.AlbumID
    JOIN Ratings r ON s.SongID = r.SongID
    WHERE r.Rating = 5
) AS song_with_ratings;

```

SongName	AlbumName
When Love Takes Over	One Love
Sine From Above	Dark & Wild
Super Rich Kids	The Fame
The Fall	A Head Full of Dreams
What Is Love	Future Nostalgia
Our First Time	Golden Hour
Ailee	Nobody Is Listening
Touch It	Over It
Happy Ending	Fine Line
Shameless	True
Héroë	Palette
People You Know	Shawn Mendes
Send My Love (To You...	The Secret
Serendipity	Love Yourself: Answer
Dreams	MUSE
Green Light	Blurryface

5. Truy vấn tên người dùng và số lượng bài hát thuộc thể loại 'Jazz' trong thư viện của họ

```

SELECT user_library.UserName, user_library.JazzSongCount
FROM (
    SELECT u.UserName, COUNT(ls.SongID) AS JazzSongCount
    FROM Users u
    JOIN Library l ON u.UserID = l.UserID
    JOIN Library_Songs ls ON l.LibraryID = ls.LibraryID
    JOIN Songs s ON ls.SongID = s.SongID
    JOIN Genres g ON s.GenreID = g.GenreID
    WHERE g.GenreName = 'Jazz'
    GROUP BY u.UserName
) AS user_library;

```

	UserName	JazzSongCount
▶	Tom Rose	1
	Vera Brooks	1
	Olivia Cyan	1
	Eva Black	1
	Kyle Davis	1
	Ivy Martinez	1
	Kelsey Carter	1
	Diana Clark	1
	George Blue	1
	Emily Grey	1
	Xander Thomas	1
	Madeline Cook	1
	Vera White	1
	Olivia Rose	1
	Liam Brown	1

6. Truy vấn danh sách các nghệ sĩ mà có ít nhất 2 bài hát được đánh giá 5 sao

```
SELECT artist_with_ratings.ArtistName
FROM (
    SELECT a.ArtistName, COUNT(DISTINCT r.SongID) AS SongCount
    FROM Artist a
    JOIN Album al ON al.ArtistID = a.ArtistID
    JOIN Songs s ON s.AlbumID = al.AlbumID
    JOIN Ratings r ON s.SongID = r.SongID
    WHERE r.Rating = 5
    GROUP BY a.ArtistName
) AS artist_with_ratings
WHERE artist_with_ratings.SongCount >= 2;
```

Result Grid	
	ArtistName
▶	Adele
	Avicii
	BTS
	Coldplay
	Drake
	Eason Chan
	Jennifer Lopez
	Jolin Tsai
	Justin Bieber
	The Weeknd
	Zayn Malik

7. Truy vấn danh sách các nghệ sĩ có ít nhất một bài hát trong album có số lượng đánh giá trung bình thấp hơn 3

```
SELECT artist_with_album.ArtistName
FROM (
    SELECT a.ArtistName, s.AlbumID
    FROM Artist a
    JOIN Album al ON al.ArtistID = a.ArtistID
```

```

JOIN Songs s ON s.AlbumID = al.AlbumID
JOIN Ratings r ON s.SongID = r.SongID
GROUP BY a.ArtistName, s.AlbumID
HAVING AVG(r.Rating) < 3
) AS artist_with_album;

```

Result Grid

ArtistName
▶ Drake
Rihanna
Alan Walker
Beyoncé
Taylor Swift
EXO
BLACKPINK
Jolin Tsai
Jolin Tsai
Rihanna
Billie Eilish
Bruno Mars
TWICE
Enrique Igl...
Ed Sheeran
Ariana Gra...

8. Tính tổng bài hát trong Playlist của từng loại thành viên

```

SELECT Member, numberOfSongsInPlaylists FROM
(SELECT u.Member, count(s.SongID) AS numberOfSongsInPlaylists FROM Users
u
JOIN Playlists p ON u.UserID = p.UserID
JOIN Playlist_Songs ps ON p.PlaylistID = ps.PlaylistID
JOIN songs s ON ps.SongID = s.SongID
GROUP BY u.Member) AS temp;

```

Result Grid

Member	numberOfSongsInPlaylists
▶ VIP	104
PREMIUM	282
NORMAL	248

E.TRANSACTIONS SỬ DỤNG ROLLBACK

1.Thêm bài hát mới (và hoàn tác) vào Album

```

start TRANSACTION;
insert into songs (SongName, Duration, PublishedDate, Language, AlbumID,
GenreID)
values ('see you again', 126, '2024-12-12', 'English', 101, 12);
select * from songs
where SongName = 'see you again' and Duration = 126;
rollback;

```

- Kết quả trước khi chạy rollback:

	SongID	SongName	Duration	PublishedDate	Language	AlbumID	GenreID
▶	1247	see you again	126	2024-12-12	English	101	12
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Thêm nghệ sĩ mới và hoàn tác

start TRANSACTION;

insert into Artist(ArtistName, Country, Style, City, DateOfBirth, Phone)
values ('Đàm Vĩnh Hưng', 'Việt Nam', 'Rock', 'Hưng Yên', '2005-11-14',
'23020390');

select * from Artist

where ArtistName = 'Đàm Vĩnh Hưng';

ROLLBACK;

- Kết quả trước khi chạy rollback:

	ArtistID	ArtistName	Country	Style	City	DateOfBirth	Phone	Email
▶	88	Đàm Vĩnh Hưng	Việt Nam	Rock	Hưng Yên	2005-11-14	23020390	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Thêm nghệ sĩ và bài hát, quay lại điểm lưu nếu không thêm được bài hát

start TRANSACTION;

savepoint add_artist;

insert into Artist (ArtistName, Country, Style, City, DateOfBirth, Phone)
values ('ali cooper', 'UK', 'Pop', 'London', '1985-06-15', '987654321');

select * from Artist

where ArtistID = last_insert_id();

insert into Songs (SongName, Duration, PublishedDate, Language, AlbumID,
GenreID)

values ('Pop Tune', 200, '2024-04-01', 'English', NULL, 1);

select * from songs

where SongID = last_insert_id();

ROLLBACK TO add_artist;

- Kết quả trước khi chạy rollback:

	ArtistID	ArtistName	Country	Style	City	DateOfBirth	Phone	Email
▶	91	ali cooper	UK	Pop	London	1985-06-15	987654321	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	SongID	SongName	Duration	PublishedDate	Language	AlbumID	GenreID
▶	1250	Pop Tune	200	2024-04-01	English	NULL	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Cập nhật xếp hạng bài hát, hoàn tác

start TRANSACTION;

update Ratings

set Rating = 5 where UserID = 10 and SongID = 65;

ROLLBACK;

- Kết quả trước khi rollback:

	RatingID	UserID	SongID	Rating	Review	CreatedDate
▶	99	10	688	5	Great song!	2011-04-21 00:00:00
*	NULL	NULL	NULL	NULL	NULL	NULL

5. Xóa đánh giá bài hát và hủy

`start TRANSACTION;`

`delete from Ratings where UserID = 193;`

`select * from Ratings where UserID = 193;`

`ROLLBACK;`

- Kết quả của UserID = 193 đánh giá trước khi chạy transaction:

	RatingID	UserID	SongID	Rating	Review	CreatedDate
▶	38	193	404	3	NULL	2012-07-12 00:00:00
*	NULL	NULL	NULL	NULL	NULL	NULL

- Kết quả sau khi chạy transaction trước rollback:

	RatingID	UserID	SongID	Rating	Review	CreatedDate
*	NULL	NULL	NULL	NULL	NULL	NULL

6. Cập nhật bài hát và thêm đánh giá bài hát nếu người dùng chưa đánh giá bài hát đó, quay lại điểm lưu đầu tiên nếu xảy ra lỗi khi đánh giá:

`start transaction;`

`savepoint add_song;`

`update Songs`

`set AlbumID = 80`

`where SongID = 225;`

`savepoint add_rating;`

`insert into Ratings (UserID, SongID, Rating, Review, CreatedDate)`

`select 2, 225, 5, 'Great Song!', '2011-12-20'`

`from dual`

`where not exists(`

`select 1 from Ratings`

`where UserID = 2 and SongID = 225`

`);`

`rollback to add_rating;`

- Kết quả trước khi chạy transaction:

	SongID	SongName	Duration	PublishedDate	Language	AlbumID	GenreID
▶	225	Pretty Little Birds	229	2017-06-09	English	43	7
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	RatingID	UserID	SongID	Rating	Review	CreatedDate
▶	2	146	225	2	Great song!	2011-12-20 00:00:00
*	NULL	NULL	NULL	NULL	NULL	NULL

- Kết quả sau khi chạy transaction (trước rollback):

✓	124	13:19:04	insert into Ratings (UserID, SongID, Rating, Review, CreatedDate) se...	0 row(s) affected Records: 0
---	-----	----------	---	------------------------------

Lệnh insert dòng trên không có tác dụng vì đã dữ liệu bị trùng

7. Cập nhật album và bài hát vào danh sách phát, quay lại điểm lưu nếu xảy ra lỗi khi thêm bài hát vào playlist (lỗi trùng lặp, lỗi dữ liệu,...)

```
start transaction;
set @songid = (select SongID from songs where SongName = 'Begin');
set @playlistid = (select PlaylistID from Playlists where PlaylistID = 32);
savepoint add_song_to_playlist;
insert into Playlist_Songs (PlaylistID, SongID)
select @playlistid, @songid
from dual
where not exists (
    select 1 from Playlist_Songs
    where PlaylistID = @playlistid and SongID = @songid
);
select * from Playlist_Songs
where PlaylistID = @playlistid and SongID = @songid;
rollback;
```

- Kết quả sau khi chạy transaction trước rollback:

	PlaylistID	SongID
▶	32	3
★	NULL	NULL

F. TRIGGER

1. Trigger để tự động cập nhật số lượng bài hát cho thành viên trong Library tự động cộng thêm 1 bài khi insert thêm song vào library/ trừ đi 1 với delete một bài trong library

DELIMITER \$\$

CREATE TRIGGER after_add_song_to_library

after insert on Library_Songs

for each row

begin

update Library

set TotalSongs = TotalSongs + 1

where LibraryID = NEW.LibraryID;

END\$\$

DELIMITER ;

- TotalSongs trước khi insert của thư viện mã số 18:

select LibraryID, TotalSongs from Library where LibraryID = 18;

	LibraryID	TotalSongs
▶	18	247
★	NULL	NULL

- TotalSongs của thư viện đó sau khi insert thêm 1 dòng:

insert into Library_Songs(LibraryID, SongID)

values(18, 34);

select LibraryID, TotalSongs from Library where LibraryID = 18;

	LibraryID	TotalSongs
▶	18	248
*	NULL	NULL

DELIMITER \$\$

CREATE TRIGGER **after_delete_song_from_library**

after delete on Library_Songs for each row

begin

update Library

set TotalSongs = TotalSongs - 1

where LibraryID = OLD.LibraryID;

end \$\$

DELIMITER ;

- Kết quả trước khi xóa một bài

select LibraryID, TotalSongs **from** Library **where** LibraryID = 18;

	LibraryID	TotalSongs
▶	18	248
*	NULL	NULL

- Kết quả sau khi xóa một bài

delete from Library_Songs

where LibraryID = 18 **and** SongID = 34;

select LibraryID, TotalSongs **from** Library **where** LibraryID = 18;

	LibraryID	TotalSongs
▶	18	247
*	NULL	NULL

2.Trigger kiểm tra bài hát có tồn tại trước khi thêm bài không

DELIMITER \$\$

create trigger **before_add_song_to_library**

before insert on Library_Songs for each row

begin

if exists (**select** 1 **from** Library_Songs

where LibraryID = NEW.LibraryID

and SongID = NEW.SongID)

then signal SQLSTATE '45000'

set message_text = 'Bài hát đã tồn tại trong thư viện';

end if;

end \$\$

DELIMITER ;

- Kết quả khi insert một bài đã có trong thư viện

insert into Library_Songs(LibraryID, SongID)

values(3, 477);

10 16:37:57 insert into Library_Songs(LibraryID, SongID) values(3, 477) Error Code: 1644. Bài hát đã tồn tại trong thư viện

3. Tự động ghi vào bảng log mỗi khi bài hát được thêm/xóa khỏi thư viện

-- Trigger ghi log khi thêm bài hát

DELIMITER \$\$

create trigger after_add_song_to_library_log

after insert on Library_Songs for each row

begin

insert into Library_Log(LibraryID, SongID, Action)

values(NEW.LibraryID, NEW.SongID, 'ADD');

end \$\$

DELIMITER ;

- Kết quả khi thực hiện hành động insert vào bảng Library_Songs, hành vi 'ADD' ngay lập tức sẽ được ghi lại tại bảng log

insert into Library_Songs(LibraryID, SongID) values(4, 577);

select * from Library_Log where LibraryID = 4 and SongID = 577;

	LogID	LibraryID	SongID	Action	ActionDate
▶	1	4	577	ADD	2024-12-14 16:44:07
*	NULL	NULL	NULL	NULL	NULL

-- Trigger ghi log khi xóa bài hát

DELIMITER \$\$

create trigger after_delete_song_from_library_log

after delete on Library_Songs for each row

begin

insert into Library_Log(LibraryID, SongID, Action)

values(OLD.LibraryID, OLD.SongID, 'REMOVE');

end \$\$

DELIMITER ;

- Kết quả khi thực hiện hành động insert vào bảng Library_Songs, hành vi 'REMOVE' ngay lập tức sẽ được ghi lại tại bảng log

delete from Library_Songs where LibraryID = 4 and SongID = 577;

select * from Library_Log where LibraryID = 4 and SongID = 577;

	LogID	LibraryID	SongID	Action	ActionDate
▶	1	4	577	ADD	2024-12-14 16:44:07
	2	4	577	REMOVE	2024-12-14 16:44:32
*	NULL	NULL	NULL	NULL	NULL

4. Trigger giới hạn số lượng bài hát của mỗi thư viện

DELIMITER \$\$

create trigger before_add_song_to_library_limit

before insert on Library_Songs for each row

begin

declare song_count int;

select TotalSongs into song_count from Library

where LibraryID = NEW.LibraryID;

```

if song_count >= 500
then
    SIGNAL SQLSTATE '45000'
    set message_text='Thư viện không thể có hơn 500 bài';
end if;
end $$
DELIMITER ;
- Kết quả khi thêm một bài hát vào thư viện đã có TotalSongs = 500
insert into Library_Songs(LibraryID, SongID)
values(last_insert_id(), 15);

```

20 16:46:09 insert into Library_Songs(LibraryID, SongID) values(last_insert_id(), 15) Error Code: 1644. Thư viện không thể có hơn 500 bài

5. Tự động chuẩn hóa định dạng Artist thành dạng đúng viết hoa chữ cái đầu tiên mỗi từ

```

DELIMITER $$
create trigger normalize_artist_name
before insert on Artist for each row
begin
    declare pos int default 1;
    declare len int default 0;
    declare result varchar(255) default "";

    set new.ArtistName = trim(lower(new.ArtistName));

    while pos <= char_length(new.ArtistName) do
        if pos = 1 or substring(new.ArtistName, pos - 1, 1) = ' ' then
            set result = concat(result, upper(substring(new.ArtistName, pos, 1)));
        else
            set result = concat(result, substring(new.ArtistName, pos, 1));
        end if;
        set pos = pos + 1;
    end while;
    set new.ArtistName = result;
end $$
DELIMITER ;
- Kết quả:
insert into Artist(ArtistName) values('KAGGGLE');
select * from Artist where ArtistID = last_insert_id();

```

	ArtistID	ArtistName	Country	Style	City	DateOfBirth	Phone	Email
▶	94	Kagggle	NULL	NULL	NULL	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. Cập nhật trạng thái thành viên: Nếu một users đã đánh giá 5 bài hát tự động nâng trạng thái Member lên PREMIUM, trên 10 là VIP

DELIMITER \$\$

```
create trigger upgrade_member_status
after insert on Ratings for each row
begin
    declare rating_count int;
    select count(*) into rating_count
    from Ratings
    where UserID = NEW.UserID;
    if rating_count >= 5 and rating_count <= 10 then
        update Users
        set Member = 'PREMIUM'
        where UserID = NEW.UserID;
    elseif rating_count > 10 then
        update Users
        set Member = 'VIP'
        where UserID = NEW.UserID;
    end if;
end $$
DELIMITER ;
```

- Trước khi thêm đánh giá cho UserID = 24

```
select count(r.Rating), r.UserID, u.Member from Ratings r
join Users u on r.UserID = u.UserID
group by r.UserID, u.Member
having r.UserID = 24;
```

	count(r.Rating)	UserID	Member
▶	4	24	NORMAL

- Sau khi thêm, User đó cập nhật lên PREMIUM

```
INSERT INTO Ratings(UserID, SongID, Rating) values
(24,12,3);
select * from Users where UserID = 24;
```

	UserID	UserName	Email	JoinedDate	Member	Address	City	Country
▶	24	Victor Blue	victorblue24@example.com	2019-07-14 00:00:00	PREMIUM	2121 Pine St	Denver	USA
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7. Tự động tính trung bình đánh giá cho những bài hát mới được thêm vào

DELIMITER \$\$

```
create trigger calculate_average_rating
after insert on Ratings for each row
begin
    update songs
    set RatingAverage = (select AVG(Rating) from Ratings
                        where SongID = NEW.SongID)
    where SongID = NEW.SongID;
```

end \$\$

DELIMITER ;

- Kết quả khi insert một bài hát mới, số sao trung bình sẽ được tính toán
`insert into Ratings (UserID, SongID, Rating) values(54, 146, 3);`
`select * from songs where SongID = 146;`

	SongID	SongName	Duration	PublishedDate	Language	AlbumID	GenreID	RatingAverage
▶	146	Bad Decisions	235	2016-05-20	English	49	1	3.00
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

8. Khi một nghệ sĩ bị xóa thì xóa những Album và Songs liên quan

DELIMITER \$\$

`create trigger cascade_artist_deletion`

`before delete on Artist for each row`

`begin`

`delete from Songs where AlbumID in (select AlbumID from Album where ArtistID = OLD.ArtistID);`

`delete from Album where ArtistID = OLD.ArtistID;`

`end $$`

DELIMITER ;

- Album của ArtistID = 6 trước khi Artist bị xóa:

	AlbumID	AlbumName	PublishedDate	ArtistID	NumberOfTracks
▶	8	Lemonade	2016-04-23	6	12
	64	Chill: Brazil (Disc 2)	2002-04-12	6	13
*	NULL	NULL	NULL	NULL	NULL

- Album của ArtistID = 6 sau khi Artist bị xóa (Album bị xóa Songs cũng bị xóa theo)

`delete from Artist where ArtistID = 6;`

`select * from Album where ArtistID = 6;`

	AlbumID	AlbumName	PublishedDate	ArtistID	NumberOfTracks
*	NULL	NULL	NULL	NULL	NULL

G.PROCEDURES

1. Dùng câu lệnh Procedures để có tên bài hát, ta có thể truy vấn đến thông tin tác giả của nó

DELIMITER \$\$

`CREATE PROCEDURE GetArtistByNameOfSong(`

`IN SongNameIs VARCHAR(100)`

`)`

`BEGIN`

`SELECT * FROM Artist a`

`join album al on al.ArtistID = a.ArtistID`

`join songs s on s.AlbumID = al.AlbumID`

`where s.SongName = SongNameIs;`

END\$\$

DELIMITER ;

- Kết quả thu được:

CALL GetArtistByNameOfSong('Back to December');

	ArtistID	ArtistName	Country	Style	City	DateOfBirth	Phone	Email	AlbumID	AlbumName	PublishedDate	ArtistID	NumberOfTracks	SongID	Son
▶	7	Drake	Canada	Hip-Hop, R&B	Toronto	1986-10-24	4567890123	drake@ovo.com	38	Views	2016-04-29	7	20	14	Back

2. Lấy danh sách bài hát trong 1 album

DELIMITER \$\$

CREATE PROCEDURE GetSongsByAlbumName(
IN AlbumNameIs VARCHAR(100)

)

BEGIN

SELECT s.SongName, s.Duration

FROM songs s

JOIN album al ON s.AlbumID = al.AlbumID

WHERE al.AlbumName = AlbumNameIs;

END \$\$

DELIMITER ;

- Kết quả:

CALL GetSongsByAlbumName('BE');

	SongName	Duration
▶	Blue & Grey	255
	Telepathy	202
	Stay	205
	Dynamite	199
	A Head Full of Dreams	237
	Birds	236
	Hymn for the Weekend	258
	Everlow	245

3. Lấy danh sách tất cả các playlists của một người dùng

DELIMITER \$\$

CREATE PROCEDURE GetPlaylistsByUser(
IN UserNameIs VARCHAR(100)

)

BEGIN

SELECT p.PlaylistName, p.CreatedDate

FROM playlists p

JOIN users u ON p.UserID = u.UserID

WHERE u.UserName = UserNameIs;

END\$\$

DELIMITER ;

- Kết quả:

CALL GetPlaylistsByUser('Maya Pink');

	PlaylistName	CreatedDate
▶	The Ultimate Rock Classics Playlist for Every Mo...	2021-02-06 00:00:00

4. Thêm một bài hát mới vào playlist

DELIMITER \$\$

```
CREATE PROCEDURE AddSongToPlaylist(  
    IN PlaylistNameIs VARCHAR(100),  
    IN SongIDs INT  
)  
BEGIN  
    DECLARE PlaylistID INT;  
    SELECT PlaylistID INTO PlaylistID  
    FROM playlists  
    WHERE PlaylistName = PlaylistNameIs;  
  
    IF PlaylistID IS NOT NULL THEN  
        INSERT INTO playlist_songs (PlaylistID, SongID)  
        VALUES (PlaylistID, SongIDs);  
    END IF;  
END$$
```

DELIMITER ;

- Kết quả:

CALL AddSongToPlaylist('Vocal Harmony Gems', 5);

	PlaylistID	PlaylistName	CreatedDate	UserID
▶	75	Vocal Harmony Gems	2022-02-21 00:00:00	1
*	NULL	NULL	NULL	NULL

5. Lấy danh sách các nghệ sĩ được theo dõi bởi một người dùng

DELIMITER \$\$

```
CREATE PROCEDURE GetFollowedArtists(  
    IN UserIDs INT  
)  
BEGIN  
    SELECT a.ArtistName, a.Country, a.Style  
    FROM artistfollow af  
    JOIN artist a ON af.ArtistID = a.ArtistID  
    WHERE af.UserID = UserIDs;  
END$$
```

DELIMITER ;

- Kết quả:

CALL GetFollowedArtists(1);

	ArtistName	Country	Style
▶	Ed Sheeran	UK	Pop, Folk

6. Đánh giá một bài hát (thêm rating vào bảng ratings)

DELIMITER \$\$

```
CREATE PROCEDURE AddRatingToSong(  
    IN UserIDs INT,  
    IN SongIDs INT,
```

```

    IN RatingValue TINYINT,
    IN ReviewText TEXT
)
BEGIN
    INSERT INTO ratings (UserID, SongID, Rating, Review, CreatedDate)
    VALUES (UserIDs, SongIDs, RatingValue, ReviewText, NOW());
END$$
DELIMITER ;
- Kết quả:
    CALL AddRatingToSong(134,2,4,NULL);
    select * from Ratings where UserID = 134 and SongID = 2;

```

	RatingID	UserID	SongID	Rating	Review	CreatedDate
▶	268	134	2	4	NULL	2024-12-14 19:43:17
*	NULL	NULL	NULL	NULL	NULL	NULL

KẾT LUẬN

Hệ thống có thể thực hiện tìm kiếm, chỉnh sửa, quản lý, cập nhật các dữ liệu liên quan đến nghệ sĩ, người dùng, bài hát,... có ràng buộc rõ ràng để giới hạn các thao tác thêm bớt, chỉnh sửa như:

- Người dùng có thể dễ dàng thêm bài hát vào playlist, cập nhật, thêm bớt đánh giá, quản lý thư viện của riêng mình, follow nghệ sĩ,... với thao tác đơn giản
- Nghệ sĩ có thể cập nhật và quản lý các album cũng như bài hát của mình, xem được đánh giá, và mức độ yêu thích của người dùng đối với bài hát, album của mình
- ...